DITA Readme map

Contents

DITA Open Toolkit

The DITA Open Toolkit is an implementation of the OASIS DITA Technical Committee's specification for DITA DTDs and Schemas. The Toolkit transforms DITA content (maps and topics) into deliverable formats, such as XHTML, PDF, Eclipse Help, HTML Help, and JavaHelp.

This set of documentation contains some basic setup and overview information for the DITA Open Toolkit. The latest information about the toolkit, including plans for upcoming or future releases, can be found at dita.xml.org: *The DITA Open Toolkit*.

Evaluating the DITA Open Toolkit (Full Easy Install)

The most common scenario for first users of the DITA Open Toolkit is to evaluate how it functions.

The most common scenario for first users of the DITA Open Toolkit is to evaluate how it functions. Previous versions of the Toolkit required a number of preliminary installation steps for components necessary both to run the base function and to show off additional output capabilities. For each release since DITA-OT 1.4.2, three versions of the Toolkit are available: the minimal package (contains core function only), standard package (minimal plus docs and demos), and a "full easy install package" that basically installs everything you need to enable quick and easy evaluation of the Toolkit. This topic deals with installing and using the full easy install package for the first time.

Note: This topic explains how to use the Full Easy Install on Windows; the three versions are also available on all other supported platforms. For a description of all packages available, see *What packages are available for the DITA Open Toolkit* at dita.xml.org.

Unzip or extract the "full_easy_install" zip file to a convenient directory, such as your c: drive's root directory. The package will create a directory such as `C:\DITA-OT\` that contains not only the usual Toolkit materials but also all the run-time components needed to run the Toolkit in a basic evaluation mode.

Browse over to this new directory and double-click on the "startcmd.bat" file in that directory. A new "command shell" window will open up, with the environment variables already set to enable the Toolkit to run within that shell.

At the command prompt (usually `C:\DITA-OT\` for this version), type "ant samples.web -f build_demo.xml". After a series of processing messages, there should be a new out\ directory in the `DITA-OT\` directory that contains a folder with the resulting HTML output in it. Congratulations on creating your first example of DITA output!

Now try the full set of transforms from a single command: "ant all -f build_demo.xml". This command will process DITA samples in the toolkit into many of the supported output types for the Toolkit. After a much longer flurry of messages, the out\ directory should have a number of folders in it, each with several deliverables produced by the Toolkit demos. If you happen to have the Microsoft HTMLHelp Workshop or the JavaHelp toolset installed for other work you have done with User Assistance, you might even get ready-to use CHM and JavaHelp output files. By comparing the outputs with the various source materials in the distribution, you can get an idea about how the processing works. That explanation is best found in the DITA Open Toolkit User Guide and at the *DITA-OT Focus Area* at http://dita.xml.org .

Demo targets

Here are the individual demo targets that you can try while evaluating the DITA Open Toolkit. Note that most are subcomponents of larger operations--you can use any of these individually. Earlier releases also contained demo targets for building the DITA 1.1 Language Specification, but that was removed in DITA-OT 1.5.2 because it was superseded by the DITA 1.2 specification.

```
all                             Build all output
clean                           Delete all output
  clean.demo                      Remove the demo output
    clean.demo.book                 Remove the book demo output
    clean.demo.elementref           Remove the Element Reference demo output
    clean.demo.enote                Remove the eNote demo output
    clean.demo.faq                  Remove the FAQ demo output
  clean.doc                       Remove the documentation output
    clean.doc.articles              Delete the articles directory in doc.
  clean.docbook                   Remove the docbook output
  clean.samples                   Remove the sample output
    clean.samples.eclipse           Remove the sample Eclipse output
    clean.samples.htmlhelp          Remove the sample HTMLHelp output
    clean.samples.javahelp          Remove the sample JavaHelp output
    clean.samples.pdf               Remove the sample PDF output
    clean.samples.web               Remove the sample web output
demo                            Build the demos
  demo.book                       Build the book demo
  demo.elementref                 Build the element reference demo
  demo.enote                      Build the eNote demo
  demo.faq                        Build the FAQ demo
doc                             Build the documentation
  doc.articles.chm                Build the articles of dita as document.
  doc.articles.pdf                Build the articles of dita as document.
  doc.articles.web                Build the articles of dita as document.
docbook                         Transform the samples to DocBook
prompt                          Prompt to build anything
samples                         Build the sample output
  samples.eclipse                 Build the samples for Eclipse
  samples.htmlhelp                Build the samples for HTMLHelp
  samples.javahelp                Build the samples for JavaHelp
  samples.pdf                     Build the samples as PDF
  samples.troff                   Build the samples as troff
  samples.web                     Build the samples for the web
```

If you do not specify a target for `build_demo.xml`, the default target is `prompt`.

You can also try your hand at modifying some of the sample scripts in the ant directory. These represent the kind of driver files that you would create for your own projects. You can easily adapt these to process your own test DITA files, for example. Call the other ant samples after this example:

```
C:\DITA-OT->ant -f samples/ant_sample/sample_xhtml.xml
```

(This is basically the same as running `ant samples.web -f build_demo.xml`, but intended for you to modify.)

You will find the output for this exercise in the `samples/ant_sample/` directory itself. You can add parameters to the `sample_xhtml.xml` file to change where your outputs end up, and also to modify the build process in other ways. See the documentation for Ant parameters to learn more about processing options.

DITA Open Toolkit Release 1.5.3

Release 1.5.3 is a maintenance release based on the final version of the DITA 1.2 standard.

Version 1.5.3 contains many enhancements, user patches, bug fixes, and significant updates to the documentation.

Release 1.5.3 was developed using a series of test builds released to the community every three weeks. Each item in the list below indicates which test build first contained the update. The eighth public build was the final build, released as the DITA-OT 1.5.3 final stable build.

General Enhancements and Changes

Base plug-ins	In earlier releases of OT, configuration parameters were hardcoded into Ant files and Java code. Starting from version 1.5.3 OT has externalized base configurations into base plug-ins in plugins

folder. Base plug-in identifiers and folder names start with `org.dita`:

- org.dita.base
- org.dita.docbook
- org.dita.eclipsecontent
- org.dita.eclipsehelp
- org.dita.htmlhelp
- org.dita.javahelp
- org.dita.odt
- org.dita.pdf
- org.dita.troff
- org.dita.wordrtf
- org.dita.xhtml

For backwards compatibility, only configuration files were moved to plug-in folders, the actual code and resource files were left in original locations.

Installations of OT may remove base plug-ins in order to remove funtionality, but the `org.dita.base` plug-in must be retained as it contains configuration for base functionality such as catalog files and preprocessing.

Plug-ing configuration changes

The plug-in configuration file `plugin.xml` has support for new syntax, where the old

```
<feature extension="foo"
 value="bar.xml" type="file"/>
```

can be written as

```
<feature extension="foo"
 file="bar.xml"/>
```

The new `file` attribute only supports a single file, not a comma separated list like the value attribute.

In previous releases multiple feature elements with the same extension ID were not supported. In release 1.5.3 multiple definitions are combined, thus

```
<feature extension="foo"
 value="bar,baz"/>
```

can also be written as

```
<feature extension="foo"
 value="bar"/>
<feature extension="foo"
 value="baz"/>
```

Plug-in extension points can be added with

```
<extension-point id="extension-id"
 name="human readable name"/>
```

Plug-ins **should** declare all extension points they support. In version 1.5.3 undeclared extension points are supported, but a warning is thrown when running integration in verbose mode. Support for undeclared extension points **may** be removed in future releases.

PDF2 changes

Support for the format attribute in PDF2 variable files has been remove as redundant. The same functionality as

```
<variable id="foo"
 format="bar">baz</variable>
```

can be implemented with e.g.

```
<variable id="foo.bar">baz</
variable>
```

PDF2 no longer logs a warning about PDF2 plug-in replacing the legacy PDF transformation type.

Support for flagging has been added.

Version of FOP that comes with Full Easy Install has been updated from 0.95 to 1.0.

Filtering configuration

List of transtypes which are considered to be print types has been moved to `integrator.properties` with the property name `print_transtypes`. In previous releases this list was hardcoded into Java code. Configuring print transtypes is currently not possible in plug-in configuration files.

Java API changes

Multiple Java classes have been changed from public non-final into package-private final. This enables clearer distinction between public and internal API, and forbits subclassing classes which have not been designed and documented for extensibility.

SourceForge Enhancements Added

1. 3177971 Improve plugin configuration file (Milestone 2)
2. 3178275 Add xsl:import extension point to PDF2 topic merge XSL (Milestone 3)
3. 3182113 Add common attribute processing to PDF2 plugin (Milestone 3)
4. 3185914 Improve integration loggin (Milestone 3)
5. 3189073 Plugin location should be available as Ant property (Milestone 3)
6. 3126848 Repository cleanup (Milestone 3)
7. 3204188 Support for defining extension point (Milestone 4)
8. 3213163 Clean PDF2 build and integration scripts (Milestone 4)
9. 3227387 Need extension to pass user param to dita.map.xhtml.toc targ (Milestone 4)
10. 3231695 Use an XML serializer object for writing XML (Milestone 4)
11. 3256796 Remove lecagy PDF code from xsl (Milestone 4)
12. 3283638 Remove format attribute support from PDF2 vars (Milestone 5)
13. 3285716 Clean up PDF2 build files (Milestone 5)
14. 3286085 Add output and temp dir params to PDF2 (Milestone 5)
15. 3293738 Use extensible pipeline task implementation (Milestone 6)
16. 3271552 ${args.xsl.pdf} as an absolut path not supported (Milestone 6)
17. 3033000 update to Apache FOP 1.0 release (Milestone 6)
18. 3213324 Separate FOP/XEP/AXF stylesheets in PDF2 (Milestone 6)

19. 3302779 Dependency extension points for PDF2 formatting (Milestone 7)
20. 3304945 Allow setting local overrides with properties file (Milestone 7)
21. 3190356 Pluginize DITA-OT base configuration files (Milestone 8)
22. 3167087 Reduce static variable usage in Java code (Milestone 8)
23. 3158929 Java clean-up (Milestone 8)
24. 3194917 Change Java API to be more final and non-public (Milestone 8)
25. 3197328 Refactor writers for cleaner XML serialization (Milestone 8)
26. 3199755 Improve log integration (Milestone 8)
27. 3296040 Refactor PDF2 build files (Milestone 8)
28. 3306146 PDF2 stylesheet refactoring (Milestone 8)
29. 3304447 Add support for selecting output format in PDF2 (Milestone 8)

SourceForge Patches Added

1. 3123507 String concat in map2plugin (Milestone 1)
2. 3110513 HTML XSLT uses complex casts (Milestone 1)
3. 3097677 Add property to reload XHTML stylesheets (Milestone 1)
4. 3106659 Added topicgroup elements to tocjs (Milestone 1)
5. 3109051 RestoreEntity duplicates functionality (Milestone 1)
6. 3107755 Configure templates with integrator properties (Milestone 1)
7. 3142967 IndexTermReader leaves tab characters in terms (Milestone 1)
8. 3140543 Add missing Commons Codec JAR into compile classpath for buildPackage.xml (Milestone 1)
9. 3087664 Clean plugin configuration parser (Milestone 1)
10. 3145258 Plug-in integrator code clean-up (Milestone 2)
11. 3147226 Use common directory layout for Junit (Milestone 2)
12. 3062765 Fix unit test file paths to be platform dependent (Milestone 2)
13. 3164523 Refactor platform Java code (Milestone 2)
14. 3160801 Improve unit test coverage (Milestone 8)
15. 3189026 Avoid strings where other types are more appropriate (Milestone 8)

SourceForge Bugs Fixed

1. 3114411 keyref links don't work for HTML Help (Milestone 1)
2. 3126578 Chunking Issues in DITA 1.5.1 (Milestone 1)
3. 3109616 More Antenna House Path Problems (Milestone 1)
4. 3155375 Incorrect way to specify recognized image extensions(Milestone 2)
5. 3157890 Navtitle Construction Does not Preserve Markup (Milestone 2)
6. 3155848 xml decl in ditaval file not closed properly (Milestone 2)
7. 3162808 Chunking remaps in-file <xref> to invalid value (Milestone 2)
8. 3164866 Upper letter estensions (Milestone 2)
9. 3165307 Add boilerplate to Java files (Milestone 2)
10. 2793836 CHM Index terms come out with extra spaces (Milestone 2)
11. 3165762 Initializer XMLReader without modifying system variables (Milestone 2)
12. 3175328 Imagemap alt text gets extra text (Milestone 2)
13. 3085106 FO: topicmerge drops id on map/topicref without href. (Milestone 2)
14. 3147328 Error in commons.xsl: getTopicrefShortdesc (Milestone 2)
15. 3130724 Error in tables.xsl: fix-relcolwidth (Milestone 2)
16. 3174906 Normalize Map and Bookmap titles for JavaHelp output (Milestone 3)
17. 3178361 Conkeyref push fails when equivalent conref push succeeds (Milestone 3)
18. 3180681 PDF2: Inconsistent template import / include. (Milestone 3)
19. 3191701 Conref Push to Same File Fails (Milestone 3)

20. 3191704 Push Replace Results in Pushed element Being removed (Milestone 3)
21. 3189883 MapLinksReader should not be namespace aware (Milestone 3)
22. 3164587 Warnings issued by Saxon 9.3.0.4 when publishing to PDF (Milestone 3)
23. 3159001 Clean unit tests (Milestone 4)
24. 3199985 @chunk : xrefs and links break (Milestone 4)
25. 3206158 Inconsistent message DOTJ038W (Milestone 4)
26. 3206373 Better handling of referenced SVG images (Milestone 4)
27. 3279539 Out of memory error from move-meta module (Milestone 5)
28. 3281108 Fallback to $locale when xml:lang value is wrong format (Milestone 5)
29. 3286679 ODT output transform deletes too many files (Milestone 6)
30. 3287609 Chunking rewrites image based on map directory (Milestone 6)
31. 3288639 Conref code improperly generalizes map domain elements (Milestone 6)
32. 3294295 PDF2 indexing and I18N fails with missing languages (Milestone 6)
33. 3294864 tocjs-demo: tocjs.ditamap is referencing a missing file (Milestone 6)
34. 3297930 PDF2: axf specific templates can't be overridden. (Milestone 7)
35. 2001271 DITA-OT documentation wants Ant 1.6.5 (Milestone 8)
36. 3136773 Incorrect version reported in log file (Milestone 8)
37. 3260746 Topichead not processed the same as title-only topic (Milestone 8)
38. 3315029 Garbled character problem in Japanese HTMLHelp (Milestone 8)
39. 3308775 Keyref map in grandparent folder fails (Milestone 8)

DITA complete release history

This document lists major changes and new features by release, going back to the beginning of the toolkit.

DITA Open Toolkit Release 1.5.3

Release 1.5.3 is a maintenance release based on the final version of the DITA 1.2 standard.

Version 1.5.3 contains many enhancements, user patches, bug fixes, and significant updates to the documentation.

Release 1.5.3 was developed using a series of test builds released to the community every three weeks. Each item in the list below indicates which test build first contained the update. The eighth public build was the final build, released as the DITA-OT 1.5.3 final stable build.

General Enhancements and Changes

Base plug-ins

In earlier releases of OT, configuration parameters were hardcoded into Ant files and Java code. Starting from version 1.5.3 OT has externalized base configurations into base plug-ins in plugins folder. Base plug-in identifiers and folder names start with `org.dita`:

- org.dita.base
- org.dita.docbook
- org.dita.eclipsecontent
- org.dita.eclipsehelp
- org.dita.htmlhelp
- org.dita.javahelp
- org.dita.odt
- org.dita.pdf
- org.dita.troff
- org.dita.wordrtf

- org.dita.xhtml

For backwards compatibility, only configuration files were moved to plug-in folders, the actual code and resource files were left in original locations.

Installations of OT may remove base plug-ins in order to remove funtionality, but the `org.dita.base` plug-in must be retained as it contains configuration for base functionality such as catalog files and preprocessing.

Plug-ing configuration changes

The plug-in configuration file `plugin.xml` has support for new syntax, where the old

```
<feature extension="foo"
 value="bar.xml" type="file"/>
```

can be written as

```
<feature extension="foo"
 file="bar.xml"/>
```

The new `file` attribute only supports a single file, not a comma separated list like the value attribute.

In previous releases multiple feature elements with the same extension ID were not supported. In release 1.5.3 multiple definitions are combined, thus

```
<feature extension="foo"
 value="bar,baz"/>
```

can also be written as

```
<feature extension="foo"
 value="bar"/>
<feature extension="foo"
 value="baz"/>
```

Plug-in extension points can be added with

```
<extension-point id="extension-id"
 name="human readable name"/>
```

Plug-ins **should** declare all extension points they support. In version 1.5.3 undeclared extension points are supported, but a warning is thrown when running integration in verbose mode. Support for undeclared extension points **may** be removed in future releases.

PDF2 changes

Support for the format attribute in PDF2 variable files has been remove as redundant. The same functionality as

```
<variable id="foo"
 format="bar">baz</variable>
```

can be implemented with e.g.

```
<variable id="foo.bar">baz</
variable>
```

PDF2 no longer logs a warning about PDF2 plug-in replacing the legacy PDF transformation type.

	Support for flagging has been added. Version of FOP that comes with Full Easy Install has been updated from 0.95 to 1.0.
Filtering configuration	List of transtypes which are considered to be print types has been moved to `integrator.properties` with the property name `print_transtypes`. In previous releases this list was hardcoded into Java code. Configuring print transtypes is currently not possible in plug-in configuration files.
Java API changes	Multiple Java classes have been changed from public non-final into package-private final. This enables clearer distinction between public and internal API, and forbits subclassing classes which have not been designed and documented for extensibility.

SourceForge Enhancements Added

1. 3177971 Improve plugin configuration file (Milestone 2)
2. 3178275 Add xsl:import extension point to PDF2 topic merge XSL (Milestone 3)
3. 3182113 Add common attribute processing to PDF2 plugin (Milestone 3)
4. 3185914 Improve integration loggin (Milestone 3)
5. 3189073 Plugin location should be available as Ant property (Milestone 3)
6. 3126848 Repository cleanup (Milestone 3)
7. 3204188 Support for defining extension point (Milestone 4)
8. 3213163 Clean PDF2 build and integration scripts (Milestone 4)
9. 3227387 Need extension to pass user param to dita.map.xhtml.toc targ (Milestone 4)
10. 3231695 Use an XML serializer object for writing XML (Milestone 4)
11. 3256796 Remove lecagy PDF code from xsl (Milestone 4)
12. 3283638 Remove format attribute support from PDF2 vars (Milestone 5)
13. 3285716 Clean up PDF2 build files (Milestone 5)
14. 3286085 Add output and temp dir params to PDF2 (Milestone 5)
15. 3293738 Use extensible pipeline task implementation (Milestone 6)
16. 3271552 ${args.xsl.pdf} as an absolut path not supported (Milestone 6)
17. 3033000 update to Apache FOP 1.0 release (Milestone 6)
18. 3213324 Separate FOP/XEP/AXF stylesheets in PDF2 (Milestone 6)
19. 3302779 Dependency extension points for PDF2 formatting (Milestone 7)
20. 3304945 Allow setting local overrides with properties file (Milestone 7)
21. 3190356 Pluginize DITA-OT base configuration files (Milestone 8)
22. 3167087 Reduce static variable usage in Java code (Milestone 8)
23. 3158929 Java clean-up (Milestone 8)
24. 3194917 Change Java API to be more final and non-public (Milestone 8)
25. 3197328 Refactor writers for cleaner XML serialization (Milestone 8)
26. 3199755 Improve log integration (Milestone 8)
27. 3296040 Refactor PDF2 build files (Milestone 8)
28. 3306146 PDF2 stylesheet refactoring (Milestone 8)
29. 3304447 Add support for selecting output format in PDF2 (Milestone 8)

SourceForge Patches Added

1. 3123507 String concat in map2plugin (Milestone 1)
2. 3110513 HTML XSLT uses complex casts (Milestone 1)
3. 3097677 Add property to reload XHTML stylesheets (Milestone 1)
4. 3106659 Added topicgroup elements to tocjs (Milestone 1)
5. 3109051 RestoreEntity duplicates functionality (Milestone 1)
6. 3107755 Configure templates with integrator properties (Milestone 1)
7. 3142967 IndexTermReader leaves tab characters in terms (Milestone 1)
8. 3140543 Add missing Commons Codec JAR into compile classpath for buildPackage.xml (Milestone 1)
9. 3087664 Clean plugin configuration parser (Milestone 1)
10. 3145258 Plug-in integrator code clean-up (Milestone 2)
11. 3147226 Use common directory layout for Junit (Milestone 2)
12. 3062765 Fix unit test file paths to be platform dependent (Milestone 2)
13. 3164523 Refactor platform Java code (Milestone 2)
14. 3160801 Improve unit test coverage (Milestone 8)
15. 3189026 Avoid strings where other types are more appropriate (Milestone 8)

SourceForge Bugs Fixed

1. 3114411 keyref links don't work for HTML Help (Milestone 1)
2. 3126578 Chunking Issues in DITA 1.5.1 (Milestone 1)
3. 3109616 More Antenna House Path Problems (Milestone 1)
4. 3155375 Incorrect way to specify recognized image extensions(Milestone 2)
5. 3157890 Navtitle Construction Does not Preserve Markup (Milestone 2)
6. 3155848 xml decl in ditaval file not closed properly (Milestone 2)
7. 3162808 Chunking remaps in-file <xref> to invalid value (Milestone 2)
8. 3164866 Upper letter estensions (Milestone 2)
9. 3165307 Add boilerplate to Java files (Milestone 2)
10. 2793836 CHM Index terms come out with extra spaces (Milestone 2)
11. 3165762 Initializer XMLReader without modifying system variables (Milestone 2)
12. 3175328 Imagemap alt text gets extra text (Milestone 2)
13. 3085106 FO: topicmerge drops id on map/topicref without href. (Milestone 2)
14. 3147328 Error in commons.xsl: getTopicrefShortdesc (Milestone 2)
15. 3130724 Error in tables.xsl: fix-relcolwidth (Milestone 2)
16. 3174906 Normalize Map and Bookmap titles for JavaHelp output (Milestone 3)
17. 3178361 Conkeyref push fails when equivalent conref push succeeds (Milestone 3)
18. 3180681 PDF2: Inconsistent template import / include. (Milestone 3)
19. 3191701 Conref Push to Same File Fails (Milestone 3)
20. 3191704 Push Replace Results in Pushed element Being removed (Milestone 3)
21. 3189883 MapLinksReader should not be namespace aware (Milestone 3)
22. 3164587 Warnings issued by Saxon 9.3.0.4 when publishing to PDF (Milestone 3)
23. 3159001 Clean unit tests (Milestone 4)
24. 3199985 @chunk : xrefs and links break (Milestone 4)
25. 3206158 Inconsistent message DOTJ038W (Milestone 4)
26. 3206373 Better handling of referenced SVG images (Milestone 4)
27. 3279539 Out of memory error from move-meta module (Milestone 5)
28. 3281108 Fallback to $locale when xml:lang value is wrong format (Milestone 5)
29. 3286679 ODT output transform deletes too many files (Milestone 6)
30. 3287609 Chunking rewrites image based on map directory (Milestone 6)
31. 3288639 Conref code improperly generalizes map domain elements (Milestone 6)

32. 3294295 PDF2 indexing and I18N fails with missing languages (Milestone 6)
33. 3294864 tocjs-demo: tocjs.ditamap is referencing a missing file (Milestone 6)
34. 3297930 PDF2: axf specific templates can't be overridden. (Milestone 7)
35. 2001271 DITA-OT documentation wants Ant 1.6.5 (Milestone 8)
36. 3136773 Incorrect version reported in log file (Milestone 8)
37. 3260746 Topichead not processed the same as title-only topic (Milestone 8)
38. 3315029 Garbled character problem in Japanese HTMLHelp (Milestone 8)
39. 3308775 Keyref map in grandparent folder fails (Milestone 8)

DITA Open Toolkit Release 1.5.2

Release 1.5.2 is a maintenance release based on the final version of the DITA 1.2 standard.

- **Release date**: December 10, 2010
- **Supports**: DITA 1.0 through 1.2
- **Download at**: *DITA-OT Latest Stable Build*
- **Which package is for me?** See *DITA-OT Packages*.

In addition to tweaks to match late changes in the standard, version 1.5.2 contains many enhancements, user patches, bug fixes, and significant updates to the documentation.

Release 1.5.2 was developed using a series of test builds released to the community every three weeks. Each item in the list below indicates which test build first contained the update. The eighth build was the final build, released as the DITA-OT 1.5.2 final stable build.

General Enhancements

1. Include final version of DITA 1.2 schemas and DTDs
2. Minor updates to DITA 1.2 support added in earlier releases, to ensure compliance with the final standard
3. Overhaul of documentation to remove outdated material
4. Reorganization of doc directory to highlight new and important info

11 SourceForge Enhancements Added

1. 2797337 Support for ODF output transform (Prototype added in version 1.5.1, updates in each 1.5.2 Milestone, transform complete in Milestone 7)
2. 3021544 Preserve DITA elements in XHTML class by default (Milestone 1)
3. 3019853 Create new "textonly" output method for use by any transform (Milestone 1)
4. 3012392 PDF transformation should allow args.xsl style override (Milestone 1)
5. 2882123 Add Ant Quick Start Guide to DITA-OT (Updated in each Milestone after 3)
6. 3086936 Add extension points for TOC output (Eclipse TOC, HTML Help TOC and Project, HTML TOC) (Milestone 6)
7. 3079610 Add current OT version to log (Milestone 6)
8. 1520909 HTML Help requires appropriate codepage (Milestone 6)
9. 3125994 Allow PDF index conf. to be overridden in Customization (Milestone 7)
10. 3125983 Create a basic glossary implementation for PDF (Milestone 7)
11. 3109395 Add parameter for Eclipse symbolic name (Milestone 8)

SourceForge Patches Added

1. 3058008 Refactor chunk module for cleaner code (Milestone 4)
2. 3067681 Add class to ordered child links (Milestone 5)
3. 3064412 Integrator fails to escape XML correctly (Milestone 5)
4. 3062765 Fix unit test file paths to be platform dependent (Milestone 5)
5. 2949860 PDF build.xml with args for JVM memory and architecture (Milestone 6)

6. 3077935 Plug-in ignore in Integrator (Milestone 6)
7. 3065050 Common logging interface (Milestone 6)
8. 3063318 ChunkModule refactoring (Milestone 6)
9. 3061100 Define AbstractPipelineInput's function (Milestone 6)
10. 3102905 Move supported image extensions to configuration file (Milestone 7)
11. 3097518 Show effective property values (Milestone 8)
12. 3101335 apiMap.mod missing from catalog (Milestone 8)

SourceForge Bugs Fixed

1. 2928582 commonTopicProcessing template prolog processing out of order (Milestone 1)
2. 2823221 version of Xalan-J inconsistent (Milestone 1)
3. 3023642 Invalid @colname generated in nested table (Milestone 1)
4. 3016739 Chunking mixes up <link> to topic in reltable (Milestone 1)
5. 3020314 Chunk output includes index terms in navtitles (Milestone 1)
6. 3020313 Chunk processor adds <topicref> before <topicmeta> (Milestone 1)
7. 3031513 Nested table processing in pdf2 (Milestone 2)
8. 3030317 Filtering doesn't work on @rev or @props attributes (Milestone 2)
9. 3028650 Replace xs:float with xs:double in Plus plugins (Milestone 2)
10. 3022847 PDF transform gives Java exceptions for spaces in dir name (Milestone 2)
11. 3032950 Scale is not correctly computed in XHTML transforms (Milestone 2)
12. 3033141 dita.xml.properties file not closed after generating (Milestone 2)
13. 3034445 "CURRENDIR" typo in plus-plugins (Checked in to CVS during Milestone 3)
14. 3034489 Remove all occurrences of <xmlcatalog> from plus-plugins (Checked in to CVS during Milestone 3)
15. 3035816 When creating .chm, .hhp-file is missing a line-break (Milestone 3)
16. 3036222 RTF transform not editable with Word 2007 (Milestone 3)
17. 3036985 Infinite recursivity in replaceString template (Milestone 3)
18. 3038941 Link with &amp; breaks in abstract (Milestone 3)
19. 3039017 Comments in PDF plugin files are confusing (Milestone 3)
20. 3058124 Toolkit Allows Unescaped URLs, doesn't handle escaped ones (Milestone 4)
21. 3056939 Conref of keyref-based xref results in xref with no href (Milestone 4)
22. 3052913 Multiple levels of keyref in map not resolved (Milestone 4)
23. 3052904 Keydef with no href causes hard failure (Milestone 4)
24. 3052156 Object with data that starts with slash breaks image copying (Milestone 4)
25. 3044861 Inappropriate warning for resource-only topic to graphic (Milestone 4)
26. 3042978 @copy-to and @chunk on topichead gives file not found (Milestone 4)
27. 3016994 The included-domains entity cannot be used in document (Milestone 4)
28. 2994593 Transformation breaks when DITA Topics contain entity refs (Milestone 4)
29. 3028894 no support for title in plugin.xml file (Milestone 4)
30. 3065853 Indent from <title> gets displayed in TOC (PDF) (Milestone 5)
31. 3065486 CURRENTFILE not aware of DITAEXT (Milestone 5)
32. 3065422 Wrong filename and filedir parameters for eclipse xsl (Milestone 5)
33. 3063533 Adjacent words get glued together using DITA to RTF (Milestone 5)
34. 3062912 Messages extension damages custom message formatting (Milestone 5)
35. 3059256 Peer links break with missing format or wrong extension (Milestone 5)
36. 2972393 Need to parameterize maxmemory and VM args for forked JVMs (Milestone 6)
37. 3060269 Problem displaying French content TOC in CHM output (Milestone 6)
38. 3038412 zh-CN file for PDF puts English strings in output (partial fix) (Milestone 6)
39. 3079676 <navtitle> contents included in PDF output (Updated so that <navtitle> in a topic will only appear when the draft parameter is set to 'yes') (Milestone 6)
40. 3004895 XHTML output for <draft-comment> should use class attribute (Milestone 6)

41. 2794487 No Easy Way to Override/Extend HTML TOC Navtitle Generation (Milestone 6)
42. 3088314 Need to clarify many error messages (Milestone 6)
43. 3095233 Shortdesc metadata missing when using abstract (Milestone 6)
44. 3081597 conkeyref accepts values in conref style (Milestone 6)
45. 3081459 fragment generation without plugin fails (Milestone 6)
46. 3073262 missing terminating quote in bundle version (Milestone 6)
47. 2832863 <group> elements in catalogs don't work for all editors (Milestone 6)
48. 3038933 Troff output drops prereq links (Milestone 7)
49. 3098975 Disable Output Escaping Should Not Be Used (Milestone 7)
50. 3102827 Allow a way to specify recognized image extensions (Milestone 7)
51. 3102219 Unexpected character code in Japanese string definition (Milestone 7)
52. 3101964 Unnecessary XML declaration in HHP and HHC (Milestone 7)
53. 3095233 Shortdesc metadata missing when using abstract (Milestone 7)
54. 3097409 PDF should skip empty columns in property tables (Milestone 7)
55. 3090803 PDF fails when chunk specified and topic appears twice (Milestone 7)
56. 3102845 Japanese character-set definition (Milestone 7)
57. 3103488 Update Saxon command line args for IDIOM PDF build.xml (Milestone 7)
58. 3026627 side-col-width variable has no effect (Milestone 7)
59. 3126007 TOC entries not properly indented in PDF (Milestone 7)
60. 3109616 Update PDF plug-in to check for latest Antenna House dirs (Milestone 8)
61. 3056040 problematic Bundle-Version test in eclipseMap (Milestone 8)
62. 3029074 Index file not generated by default for Eclipse Help (Milestone 8)
63. 3086552 XMLReader.parse does not correctly receive the XML system ID (Milestone 8)
64. 3110418 Duplicate @colname generated for entry (Milestone 8)
65. 3114353 Java sun.* packages should not be used (Milestone 8)

DITA OT Release 1.5.1

Release 1.5.1 is a maintenance release based on Committee Draft 01 of the DITA 1.2 standard.

- **Release date**: June 18, 2010
- **Supports**: DITA 1.0 through 1.2 (Committee Draft 02 level)
- **Download at**: *DITA-OT Latest Stable Build*
- **Which package is for me?** See *DITA-OT Packages*.

This is the same version of the standard used for the DITA 1.2 Public Review. Release 1.5.1 contains many fixes and minor enhancements. It also includes a preview of a new output transform to the Open Document Format; this transform will be completed in a later release.

Release 1.5.1 was developed using a series of test builds released to the community every three weeks. Each item in the list below indicates which test build first contained the update. There were seven total test builds.

General Enhancements

1. Update to latest copy of DITA 1.2 Draft DTDs and Schemas (last update in Milestone 5)

14 SourceForge Enhancements Added

1. 2797337 Support for ODF output transform (first prototype available in Milestone 2, further updates in each milestone)
2. 2703335 Reduce duplicated code in XHTML <note> processing (Milestone 3)
3. 2976463 Provide finer grained control of links in PDF (include reltable and in-topic links, without parent/child links) (Milestone 4)
4. 2971536 New Java options for existing Ant parameters (Milestone 4)

5. 2979084 Obey the "args.draft" parameter (Milestone 5)
6. 2990783 allow caller-provided IndexTermCollection (Milestone 6 contains the core code updates; M7 contains the full enhancement)
7. 3001750 Shortdesc from map should override link description in PDF (Milestone 7)
8. 3004305 Notes with type="warning" need styling / localization in XHTML (Milestone 7)
9. 3004859 "eclipsecontent" transform should drop debug info (Milestone 7)
10. 2892706 Don't delete the FO file (new option to preserve topic.fo) (Milestone 7)
11. 2928584 Add general model for end-of-topic processing in PDF (Milestone 7)
12. 3006675 Support new DITA 1.2 <stepsection> element in PDF (Milestone 7)
13. 3006847 Add generated task headers to PDF (using the option that works for XHTML in DITA-OT 1.5) (Milestone 7)
14. 2987769 Add support for Eclipse Help index redirects (Milestone 7)

2 SourceForge Patches Added

1. 2914475 Use Xerces Grammar Pool to Improve Performance (Milestone 1)
2. 2991688 Refine package build Ant (Milestone 6)

46 SourceForge Bugs Fixed

1. 2916469 @locktitle not respected by preprocessor (Milestone 1)
2. 2917809 need empty lib/saxon directory for minimum and standard pkg. (Milestone 1)
3. 2925636 Build fails when excluded section contains a table (Milestone 1)
4. 2926417 Absolute file: URLs are not resolved. (Milestone 1)
5. 2930109 Move meta module pushes content into peer topic. (Milestone 1)
6. 1852808 args.css requires dummy file. (Milestone 1)
7. 2952956 Imagemap processing drops outputclass from image (Milestone 3)
8. 2953706 Minor improvements to "garage" samples (Milestone 3)
9. 2961909 /onlytopicinmap does not respect conref (Milestone 3)
10. 2957456 conkeyref breaks when topic is in subdir (Milestone 3)
11. 2962813 stepsection throws off numbering in links to steps (Milestone 3)
12. 2959588 Template Processor Doesn't handle XSLT atts that require ' (Milestone 3)
13. 2914574 plus-htmlhelp-alias-map: using same extension point twice (Milestone 3)
14. 2957938 coderef not working everytime (Milestone 3)
15. 2962781 html documentation out of date (Milestone 3)
16. 2952956 refactored XSL code in ut-d.xsl (Milestone 3)
17. 2954819 NullPointer while processing simple BookMap (Milestone 3)
18. 2954154 Updated default version from 1.0 to 1.0.0 (Milestone 3)
19. 2970471 XSLFO test for @compact wrong (Milestone 4 for PDF, Milestone 5 for LegacyPDF)
20. 2972043 Setting onlytopicinmap causes a blank imagelist (Milestone 4)
21. 2974667 Integrator adds spaces into XML Catalog entries (Milestone 4)
22. 2986492 Duplicate parameter in XHTML code (Milestone 5)
23. 2982485 Cannot read a document that was written during the same transform (Milestone 5)
24. 2981216 <tm> @tmclass requires IBM-specific values (Milestone 5)
25. 2979361 Java stack traces in OT log (Milestone 5)
26. 2979328 Output parameters info at INFO level (Milestone 5)
27. 2978858 keyref processing doesn't respect basedir (Milestone 5)
28. 2990162 Conref to elements in same DITA file throw parsing errors (Milestone 6)
29. 3000677 msgph and systemoutput should use <samp> instead of <tt> (Milestone 7)
30. 3004220 <tm> elements dropped when keyref text resolved (Milestone 7)
31. 2987322 Navtitle attribute of glossarylist breaks PDF (Milestone 7)

32. 2916474 Inappropriate match on mapgroup/topichead in PDF code (Milestone 7)
33. 2916408 Topichead with <navtitle> not processed same as @navtitle in PDF (Milestone 7)
34. 3006443 CSS for prereq links indents prereq element (Milestone 7)
35. 2607892 (Plus Plug-in) plus-allhtml-encoding: map x-windows-950 to Big5 (Milestone 7)
36. 2385466 Handle @font-family="inherit" (Milestone 7)
37. 2928540 Shortdesc should align with body in PDF (use topic__shortdesc) (Milestone 7)
38. 1839827 PDF does not properly process <xref> to <li> elements (Milestone 7)
39. 2521819 PDF topic title widowed due to fo:marker placed in separate block (Milestone 7)
40. 1385654 docbook/topic2db.xsl - better linking support (Milestone 7)
41. 3004550 Conkeyref does not work if key referenced is not in same folder (Milestone 7)
42. 3004060 keyref/id not producing link in PDF output (Milestone 7)
43. 3001705 conkeyref doesn't work across ditamaps (Milestone 7)
44. 3000604 Legacy PDF: empty @column-number causes errors (Milestone 7)
45. 3013079 Keyref handling does not respect scope="external" (Final build)
46. 3005748 XHTML: Topics w/ @print set to "printonly" are included (Final build)

DITA OT release 1.5

Release 1.5 is a major release based on the new DITA 1.2 draft standard.

It contains full support for DITA 1.2 draft as defined in the fall of 2009 (prior to public review).

In addition to DITA 1.2 support, release 1.5 contains the following updates, which are available in Milestone 21 or earlier of the DITA-OT test builds:

General Enhancements

1. New parameter to generate output for only the topics referenced in a map
2. Use fileparameter in Ant 1.7 to replace processing instruction in intermediate dita files
3. Determine the version of DITA-OT via the class org.dita.dost.util.Version
4. Remove duplicate ids in PDF topicmerge
5. Move Notices before the TOC, suppress in the TOC and suppress the second copy after the TOC
6. Include @scope="peer" condition when creating rel-links.

SourceForge Enhancements Added

1. 2859612 Add support for Serbian (Latin)
2. 2845278 Add Ant parameters for XSLT parameters
3. 2824371 Support Hindi and Urdu for XHTML output
4. 2790755 Process in unique temp directories (Designed to allow multiple builds to take place at once)
5. 2780998 startcmd.sh: Run the user's shell, not sh
6. 2698921 Add a way to set attributes on <body>
7. 2120219 Add PDF to the supported image
8. 2002857 homeID in javahelp .hs file is not set
9. 1725284 add support for headings for sections within task (XHTML only)
10. 1623246 Support RFC4646 language tags in Index modules
11. 1367897 Xref content generation enforces formatting choices
12. 2882109 Convert named PDF templates to mode template
13. 2883406 Add extension point for modifying PDF link text
14. 2882870 Add parameter to control PDF Chapter style
15. 2882103 Provide hook for specializations to add custom headers
16. 2882072 Add parameter to control PDF bookmarks (default collapsed)

93 SourceForge Bugs Fixed

1. 2860433 Keyref on <image> fails to resolve
2. 2860199 Chunk to-content in submap resulting in missing output
3. 2860168 Suppress data and data-about in PDF output
4. 2857167 conrefImpl.xsl 1.6.4.13 has duplicate variable decls
5. 2856742 Problem of keys attribute in nested topicref elements
6. 2854546 Peer xref leads to invalid destination error in PDF
7. 2849200 Style on ditaval prop or revprop is ignored
8. 2846111 Unnecessary text generated for external xref
9. 2845598 1.5-M18-demo/fo/Customization pdf.formatter
10. 2842753 catalog-dita.xml has invalid entries
11. 2839035 Chunk code cannot parse xref with &
12. 2832696 Move meta module pushes content into non-DITA file
13. 2829350 Build_demo.xml places files in incorrect directory
14. 2826143 Move meta module discards link text
15. 2824907 1.5 M17 Bug - FileNotFoundException export.xml eclipsehelp
16. 2819853 missing nested indexterm after 3rd
17. 2815492 PDF: keep-with-next on shortdesc
18. 2815485 Prolog indexterm not processed properly for PDF
19. 2813082 Eclipse help MANIFEST.MF filename wrong case, needs CRLF
20. 2811980 pdf2: japanese translate for generated page number of a ref
21. 2811358 Incorrect entry@colname in middle file at PDF generation
22. 2804442 Bad param name in prereq-fmt template's call to sect-heading
23. 2799543 Child links in HTML Help output are wrong
24. 2797030 Chunking fails with bookmap
25. 2796614 Leading slash breaks PLUGINS_ROOT usage for Eclipse help
26. 2791696 reltable DITA 1.2 (#12048)
27. 2791345 Topicmeta searchtitles in map not used in topic
28. 2791278 Keyref Resolution Fails for Non-Descendant Topics
29. 2790807 Demo code should use new PDF output
30. 2788069 Topicmerge does not handle copy-to properly
31. 2788069 Topicmerge does not handle copy-to properly
32. 2782503 Extra space before bullet list in xhtml output DITA-OT 1.4.3
33. 2774128 startcmd.sh is encoded with DOS line endings in v1.5 M13
34. 2759964 HTML outputs filters conditional topics poorly
35. 2759964 HTML outputs filters conditional topics poorly
36. 2748371 Revision + ID gives XSL error
37. 2739236 plus-allhtml-svgobject handles <alt> badly
38. 2724090 XHTML: conreffed by-reference footnotes break w/ chunking
39. 2723928 gen-toc template not matching properly
40. 2723715 Itemgroup sets @id when element is not open
41. 2712074 XHTML: chunk=to-content on map breaks by-reference footnotes
42. 2706725 Single quote inserted in empty table cells in FO
43. 2696229 FO plugin code incompatible with Saxon 9 (OT1.4.3 default)
44. 2696191 Java libraries missing from 1.4.3 distributions
45. 2647292 HTML stylesheets should style <wintitle> and <filepath>
46. 2629271 maplink: should not modify href if scope=external
47. 2629256 mapref: should not modify href if scope=external
48. 2573681 Move link module does extra processing

49. 2547437 zh_TW.properties and zh_CN.properties contents are mixed up
50. 2418932 toc attribute does not work for references to ditamaps
51. 2317681 Extra files generated when many topics are chunked to one
52. 2317581 inline formatting in shortdesc
53. 1931457 Need to identify transtype during XHTML output step
54. 1629094 docbook creates empty simplelists
55. 1628936 transtype=docbook does not handle publisher and copyright element
56. 2849078 Problems using keyrefs with DITA OT 1.5 M19 - ID: 2849078
57. 2875373 tm in linktext is dropped
58. 2870935 keyref within topics ignore @copy-to attributes
59. 2873560 SWF flash not defined as resources to be copied
60. 2872954 Conref push not working at map level
61. 2871009 Temp directory leaves behind single directory and file
62. 2866322 Generated links have bad URLs
63. 2873654 PDF missing rules for new DITA 1.2 content elements
64. 2872988 Bad relcolwidth crashes PDF output
65. 2866204 Topicref with keyref fails to produce output
66. 2878446 Issue with ampersand in xrefs in attributes
67. 1629096 docbook creates invalid varlistentry
68. 2871326 Cannot use different TOC titles for same topic in PDF
69. 1880097 PDF2 ingnores contents of <xref> elements
70. 1815571 Invalid property in fo:table-body
71. 2879171 Shortdesc & Abstract formatting is incorrect for PDF
72. 2882085 Obsolete code in PDF plug-in should be removed
73. 2661418 make the TOC in pdf2 output link to topics in pdf
74. 2871017 eclipsehelp plugin.xml invalid in 1.5M20
75. 2887331 chunk="to-content" on a sub-map causes path error
76. 2891736 indexterm in topicref level are copied into topic/prolog
77. 2893316 This is a bug in the integrator.xml
78. 2893493 ${clean.temp} is not set to true per documentation
79. 2893664 ampersand entity with conref actions causes build to fail
80. 2893924 htmlhelp transtype requires video plugin
81. 1628937 Rename supportingboth.ditaand.xmlinaditamap.dita
82. 1771123 Inappropriate Message 018 On type= value
83. 1819660 Eclipse Help issue producing toc file.
84. 1848313 ditaval file sent to XSLT not URL
85. 1897542 Chunk to-content may need to rewrite topic IDs
86. 2875946 PDF: table @pgwide semantics not implemented
87. 2893745 missing fop's file/directory in standard package
88. 2900047 map2htmtoc.xsl doesn't allow for extension of root template
89. 2900417 html transform does not use image/@scale properly
90. 2906957 Update Eclipse plugin DTDs and packaging
91. 2796964 Use xml:lang for generated text in PDF
92. 2860596 PullPrologIndexTerms selection too wide, Removed spurious pullPrologIndexTerms in section/title
93. 1852733 Image "scale" attribute is ignored

4 SourceForge Patches Added

1. 2790337 Add extension points to related-links code
2. 2778178 Flagging code duplicates XSL

3. 2715750 plus-transtype-htmlhelp: support alternate compilers
4. 2804311 Feature value trim test bug

DITA OT release 1.4.3

Release 1.4.3 is a maintenance release and is intended to be the last released based fully on DITA 1.1.

It contains a significant amount of preview function for the upcoming DITA 1.2 standard, but the DITA 1.2 document types must be specifically referenced in order to use this capability.

Major Updates

1. The PDF codebase switched from the older deprecated code to the plugin previously known as PDF2. Running a build with the transform type PDF or PDF2 will now run the new code. To continue using the older code, set your transform type to "legacypdf".
2. The PDF plugin now runs with FOP, as well as with the latest version of Saxon.
3. The Full Easy Install package now ships with Saxon 9B instead of with Xalan, and code in the DITA-OT package has been updated to work with Saxon 9B.
4. The Full Easy Install package updates the versions of many open source libraries (including Ant and FOP)
5. RFE 1982567 Allow spaces in DITA file names
6. Preview function for the majority of the upcoming OASIS DITA 1.2 standard, including everything from milestone builds one through twelve of DITA-OT 1.5

10 SF Enhancements Added

1. 1982567 Allow spaces in DITA file names
2. 2631145 dita2htmlImpl.xsl should not define doctype
3. 2562718 Rename the ant directory to avoid naming collisions with ant
4. 2314086 Let maplink generate <linklist> elements
5. 2258659 Extend integrator to allow addition of messages.
6. 2117337 Customize Integrator to integrate custom directories
7. 2074933 Make index output for new transtypes more extensible
8. 1995667 Add plugin extension points to preprocess pipeline
9. 1972537 Let users specify reloadstylesheet on xslt tasks
10. 1803199 Allow ability to include class ancestry in XHTML class attribute

4 SourceForge Patches Added

1. 1996733 Structure Ant <pipeline> better
2. 1796007 Refactor of related links HTML for specialized processing
3. 1947566 pdf2: allow XEP to be installed elsewhere
4. 2477370 Refactor of conrefImpl.xsl

31 SF Bugs Fixed

1. 2008294 End flag does not work in ditaval
2. 2078563 Keyref not working for external links
3. 2027170 maprefImpl.xsl removes part of class attr from topichead
4. 2027058 Topichead element stops map processing in ChunkMapReader
5. 2001268 DITA-OT documentation wants JDK 1.4
6. 2008317 Chunking one topic from a large file hogs memory
7. 2219479 <prodinfo> subelement associations are lost in HTML output
8. 2340727 No link in TOC to topics in ditabase
9. 2317627 Chunking map to create one file should use map file name

10. 2143078 Relcolwidth not respected in choicetable
11. 1995223 bookmap: bookmeta->bookrights information is not generated
12. 1990167 reltable doesn't generate external links correctly
13. 1953553 FO 1.4.2 plugin: Inconsistent display of "on page" in xref
14. 1951879 Link generation is disabled for bookmaps
15. 1997171 topic body suppressed in chapter topics
16. 2417980 Toolkit should support both versions of Serbian lang code
17. 1947817 Extra topics appear in PDF output when using reltables
18. 2004588 Image alt text needs space normalized
19. 1954463 Conditional processing multiple PDFs in the same JVM
20. 2061301 Useless import of xslt4j stops dost.jar being built
21. 2317659 Chunking "by-topic" does not work
22. 1955211 Footnotes-by-reference rendered naively
23. 2010062 Spaces dropped in indexterm that has keyword
24. 2607496 USEINDEX=no breaks HTML Help HHP
25. 2570118 <xref> without href drops content
26. 2414891 Map contains wrong reference after chunking
27. 2010092 example or section title can be orphaned in PDF2
28. 1999117 Ubuntu 8.04 | startcmd.sh doesn't work
29. 2020317 Many duplicated id in xxx_MERGED.xml in PDF2 output.
30. 2614006 Conref processing ignores -dita-use-conref-target
31. 1727863 <groupchoice> has an extra "|" character

DITA OT release 1.4.2.1

Release 1.4.2.1 is a full build to provide an urgent fix to fix the following critical problem which users found in release 1.4.2.

SF Bugs Fixed

1. SF Bug 1956231 OSGI Manifest for eclipsehelp transtype contains spaces.

For this fix, we have updated xsl/map2pluginImpl.xsl to prevent generating extra spaces at the head of every line in OSGI manifest file of eclipse help output.

DITA OT release 1.4.2

Release 1.4.2 is a maintenance release to fix defects and make patches based on release 1.4.1.

Release 1.4.2 comes in three versions:

1. Minimal package contains only the core processing code.
2. Standard package contains the core processing code, plus demos, documentation, and samples.
3. Full "easy install" package contains the standard package plus external libraries useful for many toolkit processes (such as Xalan and ANT), plus a batch file to setup a local environment.

39 SF Bugs Fixed

1. 1945824 Index-see works for "ru-ru" but not for "ru"
2. 1944245 Null pointer error with complex filename
3. 1923519 Conrefs in nested, conreffed topics don't work
4. 1911285 files.txt is not up-to-date
5. 1906954 Constants.ANT_INVOKER_EXT_PARAM_OUTPUTDIR resulted in null ...
6. 1903830 Error when collection-type=sequence in map
7. 1903626 Topicref to Glossentry With Topicmeta Throws Null Pointer

8. 1900907 Documentation of generateouter command-line param is incorrect
9. 1900427 TIFF file format not a supported type
10. 1898810 Problem running ant with DITA-OT in path with Latin char
11. 1897358 Compiling CHM's in sequence results in errant index entries
12. 1894561 wordrtf not correctly handling p inside li
13. 1893234 Java TopicMerge removes processing instruction
14. 1868423 Null pointer exception when a PI is at the end of the file
15. 1857405 chunk processing fails when no section element
16. 1855047 startup.sh fails under OS X
17. 1849346 FO file generated from DITA MAP not valid
18. 1843652 Image referenced in map is not found, topicmerge breaks
19. 1843583 Extra bullet in TOC for topicref with no href or navtitle
20. 1839765 index-sort-as not used, content appears in index
21. 1908306 Index entry for external resource is ignored
22. 1908293 HTML Help index contains extra anchor
23. 1900916 Pointer to CSS is Incorrect in index.html
24. 1898451 HTML titles should be space-normalized for CHM consumption
25. 1898228 Table desc not being processed
26. 1897551 maplink is unawareof chunk to-content
27. 1893461 map file href handling
28. 1889918 Index link goes with wrong entry
29. 1883907 IndexTermReader class doesn't handle specialized titles
30. 1873401 XHTML: colsep in last column when @frame=none
31. 1872434 _merge.xml missing metadata
32. 1868047 htmlhelp path in demo ant script is fixed
33. 1864247 PIs missing from ditamaps in temp dir
34. 1857282 path to css output is not correct
35. 1848355 gen-list wants class on <foreign> descendants
36. 1843693 Bad XREF syntax gives confusing message about unique_193
37. 1841175 Need to clean up doc/ directory (remove invalid items)
38. 1832800 Empty end-range indexterm causes eclipse error
39. 1606387 Shortdesc & Abstract formatting is incorrect for XHTML

3 SF Patches Added

1. 1930220 Simplify flag templates
2. 1876118 Add plug-in support for string resource-files
3. 1818318 Path to HTML Help compiler on x64 Windows

6 SF Enhancements Added

1. 1855523 Pass dost.class.path to XSLT tasks
2. 1827322 Let plugins add their own template files
3. 1825843 Let plugins add dependencies to Ant targets
4. 1824466 Subclass ImportAction
5. 1782256 Let plugins not have to choose to go in "demo" or "plugins"
6. 1859421 Add plug-in support for string resource-files

DITA OT release 1.4.1

Release 1.4.1 is a maintenance release to fix defects and make patches based on release 1.4.

23 SF Bugs Fixed

1. 1833801 Infinite loop in MapMetaReader
2. 1833796 move-meta-entries creates invalid XML
3. 1827055 Dita 1.4 move metadata method failing
4. 1819663 XHTML processing add &nbsp; in output files.
5. 1815155 Using xref moves output directory
6. 1807808 Java TopicMerge calling XSLT transformer with URL not file
7. 1806728 Merge doesn't normalize filenames
8. 1806130 chunk module wraps long lines
9. 1806081 <dita> without class attribute triggers warning
10. 1803190 XHTML: processing <xref> to <a title="">
11. 1803183 XHTML: <b> and <xref> within <pre>
12. 1796207 topicmeta in ditamap causes build failure
13. 1782109 Title input to Help Compiler invalid for taskbook example
14. 1779066 [DOTX031E] Errors
15. 1770571 Chunk "to-content" on map not implemented
16. 1732678 Map without DOCTYPE declaration produces odd error
17. 1675195 No Error Location for Titleless Topic
18. 1639672 The Toolkit does not properly support valid xml:lang values.
19. 1639344 Xref : topicpull : the spectitle not used as linktext
20. 1628937 Rename supportingboth.ditaand.xmlinaditamap.dita
21. 1584187 Bookmap 1.1: <title> element breaks topicmerge
22. 1563093 Difficult to find location of error
23. 1505172 foimgext Considered Harmful

5 SF Patches Added

1. 1741302 Prevent indexterm crash with two-letter language codes
2. 1630214 HTML Help HHP generator: Language tag
3. 1498936 Failure when moving links with embedded mathml
4. 1481586 CSS for ditamap-to-HTML TOC
5. 1457541 xref to elements fails within topics in PDF

5 SF RFE Added

1. 1764910 Allow greater control over the output directory
2. 1764905 Allow option to build only topics listed in the map
3. 1725280 Improve error reporting in general
4. 1686939 Make dita.list into an XML file
5. 1676947 Integration points for passing params to XSL

DITA OT release 1.4

Release 1.4 is a major release to add new functions, fulfill new requirements, make some function enhancements and fix bugs over release 1.3.1. Available since August 1, 2007

The DITA-OT Release 1.4 contains full support for the OASIS DITA 1.1 standard. This completes the preliminary support added in the 1.3 and 1.3.1 versions of the toolkit. New and improved items for 1.1 are listed under [Improvements] below. Support for the new bookmap standard is available in the latest version of the FO plug-in, which uses the "pdf2" transform type; it will be released together with or soon after the release of DITA-OT 1.4. The deprecated "pdf" transform type has not been updated for the new bookmap. Together with DITA 1.1 support, the toolkit development team has improved error reporting so that build failures are more accurately reported at the end of the build. Error handling will continue to improve in future releases. Release 1.4 comes in two versions. The full version contains several external packages that are useful or critical to running the toolkit, such as Xalan and the XML Catalog resolver. The smaller package contains only core toolkit code. NOTE ABOUT DEPRECATED CODE: changes for the new DITAVAL standard required a change to code in dita2htmlImpl.xsl. The "flagit" named template is deprecated and will not work with the new ditaval format. Overrides to this step should be updated to use "start-flagit" and "end-flagit". The flagit template will continue to work with the old ditaval but will generate a warning for each call.

Changes

1. Release 1.4 improves the processing of DITA documents using XML Schemas. One was able to process these type of documents in Release 1.3.1 but it meant that the schema location had to have the ablsolute location of the schema in order for the Toolkit properly.

   DITA 1.1 introduces the use of URNs to normatively identify the schemas used for validation. The URNs have the following desing pattern "urn:oasis:names:tc:dita:xsd:<schemaDocument>:1.1". You should use these in as the value for the attribute xsi:noNamespaceSchemaLocation.

13 Improvements

1. Support <title> in map
2. Ignore Index-base in default processing
3. Retrieve the link text from abstract element.
4. Format shortdesc in abstract appropriately
5. Add standard code to allow overrides to easily process generalized version of unknown and foreign element
6. Support @dir on every element
7. Refactor mapref resolution
8. Support generalization and re-specialization of unknown/foreign elements
9. Replace Move Index module with new Move Metadata module
10. New DITAVAL standard support
11. New chunk attribute support
12. Support XML Schema validated instance document processing using XML Catalogs

17 SF Bugs Fixed

1. 1700561 Null Pointer Exception on Missing domain= Attribute
2. 1733264 pretty.xsl is broken
3. 1619074 table in step screws up following steps for HTML generation
4. 1728700 GenMapAndTopicList keeps filtering when called a second time
5. 1732562 DitaWriter.java can duplicate @xtrf and @xtrc
6. 1733108 Update Bookmap sample files to DITA 1.1
7. 1706263 Conrefing from a map to topic is not working properly
8. 1677620 Non-DITA file is treated as DITA in pre-process
9. 1717471 Links show up more than once
10. 1712543 gen-list-without-flagging : NullPointerException
11. 1652892 Invalid hdr/ftr arg value causes build failure
12. 1647950 PIs in DITA source are dropped in the processing pipeline
13. 1644559 Force Toolkit to use private catalog to allow schemas to work properly
14. 1642138 Move javamerge target out of build_template.xml
15. 1643155 Map TOC is HTML even for transtype="xhtml"
16. 1637564 topicpull breaks specializations of xref

17. 1676968 Plugins adding to classpath break when basedir != dita.dir

DITA OT release 1.3.1

Release 1.3.1 is a maintenance release to fix defects and make patches based on release 1.3.

15 SF Bugs Fixed

1. SF Bug 1385642 docbook/topic2db.xsl - shortdesc
2. SF Bug 1528638 wordrtf does not correctly number steps
3. SF Bug 1562518 Flag is confusing when a list is mixed with text
4. SF Bug 1563665 Should use CSS to honor rowsep and colsep in table entries
5. SF Bug 1567117 Xref to footnote is not resolved correctly
6. SF Bug 1569671 <reltable> in nested map creates bogus TOC entries
7. SF Bug 1573996 Plugins do not work in plugins directory
8. SF Bug 1574011 Spaces in a file name prevent XHTML output
9. SF Bug 1584186 Bookmap 1.1: <title> element duplicated in mappull
10. SF Bug 1588039 Conref domain checking is sub-par
11. SF Bug 1588624 OT v1.3 map2hhc.xsl error
12. SF Bug 1597444 Java topicmerge breaks when text contains less-than
13. SF Bug 1597473 Nothing references common.css
14. SF Bug 1598109 Java topicmerge does not rewrite image/@href
15. SF Bug 1598230 jhindexer of JavaHelp breaks Search Index for DITAOT content

DITA OT release 1.3

OASIS DITA 1.1 support

Things to know about OASIS DITA 1.1 support in this release:

1. DITA-OT 1.3 provides preliminary processing support for the upcoming OASIS DITA 1.1 specification (see *http://wiki.oasis-open.org/dita/Roadmap_for_DITA_development*). Because the proposed OASIS DITA 1.1 DTDs and Schemas are fully backwards compatible with the latest DITA 1.0.1 DTDs and Schemas, the 1.3 Toolkit provides the proposed 1.1 materials as the default DTDs for processing. The XML Catalog resolution maps any references for DITA 1.0 doctypes to the 1.1 DTDs, for example. All processing ordinarily dependent on the 1.0 definition continues to work as usual, and any documents that make use of the newer 1.1-based elements or attributes will be supported with specific new processing function (such as base support for the new <data> element). *Documents created with the proposed OASIS DITA 1.1 DTDs are the only ones ever likely to have features that invoke the specific new 1.1-based processing support.*

   Important: Because this support is based on a yet-to-be-approved version of the proposed OASIS DITA 1.1 specification, if you choose to investigate any1.1-based function, be aware that the 1.1 implementation in this version of the Toolkit is preliminary and very much forward-looking. Upon final approval of the DITA 1.1 standard, Toolkit developers will, of course, review our implementation to make certain that it conforms to the defined level of reference implementation.
2. Related to the DITA 1.1 preliminary implementation, the much-discussed bookmap updates for DITA 1.1 will be provided as override capabilities for the FO plugin (Idiom's donation). Note that:

   - The FO demo transform code at the 1.2.2 level is still included in the DITA 1.3 package, but is now deprecated.
   - To get the FO updates for 1.3, grab the FO plug-in at its next update, which should be shortly after the 1.3 core Toolkit code is released.
   - The updated FO plug-in will be usable with FOP as well as with XEP.

Changes

The DITA Open Toolkit team understands the need for stability in essential APIs in the Toolit. This verson of the toolkit provides some strategic updates that correct some long-overdue faults in the original implementation. Necessarily, there are some changes to note:

1. Change to build.xml: To make the DITA processing environment more like other Ant-driven build environments, the original build.xml has been renamed as build_demo.xml. The current build.xml in this release is now the normal ANT script entrance for starting a transformation. If you have created Ant tasks that tried to work around the former build.xml architecture, those might need to be revised to take advantage of the separated function.
2. Change to command line invocations: The "Ant refactoring" exercise for this release has changed some previously documented Ant calls for running demos. This change enables better use of the Ant modules for power users who need to integrate the Toolkit into programming build environments such as Eclipse, but the change affects some documentation. This is a permanent change that should remain stable from now on. Wherever you see an older instruction like "c:\dita-ot>ant all", you now need to indicate the component that contains the demos, so you would type "c:\dita-ot>ant all -f build_demo.xml".
3. Separation of demo targets from formal component targets: Another effect of the Ant refactoring is that the internal programming targets will now be displayed when you type "ant -p". To see both those programmings targets and the demos that are part of this component, type "c:\dita-ot>ant -p -f build_demo.xml". To run just one of the demos that you see in the resulting list, dita.faq for example, type "c:\dita-ot>ant dita.faq -f build_demo.xml".
4. Classpath update to enable catalog resolver: This release now includes the Apache catalog resolver for improved lookup of DTDs by any of the Toolkit components. The fullpackage version of the Toolkit sets up these variables for each session. For the regular (smaller) version of the Toolkit, you need to include lib and lib\resource \resolver.jar into your classpath. For example if your CLASSPATH is like:

   ```
   c:\dita-ot\lib\dost.jar
   ```

   you need to change it to:

   ```
   c:\dita-ot\lib;c:\dita-ot\lib\dost.jar;c:\dita-ot\lib\resolver.jar
   ```

   At any time, the full version can be used like a normal installation as long as you update the system variables either in the evironment settings or in a batch file that sets up the shell environment.
5. License bundling: To reduce the duplication of builds on Sourceforge in which the only difference was the license provided in each, both the Apache and CPL licenses are included in root directory of the Toolkit. Use the one that applies to your situation.
6. Two install options: Two download versions are now offered. The smaller one is for updating existing installations or for reuse in embedded applications that already provide the other processing components--business as usual. A new package with "fullpackage" in the name now incorporates the essential processing modules to create a processing environment for new users and evaluators that requires nothing more than to unzip the file into an appropriate directory and then click on a "start" batch file. A new document in its root directory (an output of doc/EvaluateOT.dita, “Evaluating the DITA Open Toolkit (fullpackage version)”) informs new users how to install and use the Toolkit for the first time.
7. Other enhancements: The public design discussions that fed into the final selection and architectures for this release are documented at the DITA Focus Area in a topic called “DITA OT 1.3 Issues tracking” (*http://dita.xml.org/node/1282*).

7 Improvements

1. Preliminary support for OASIS DITA 1.1
2. Support ICU in index sorting
3. Integrate with Eclipse
4. Refactor Ant script for easy override
5. Topicmerge reimplementation in JAVA
6. Enable XML Catalog Resolver
7. Full package distribution (was GUI/usability)

21 SourceForge Bugs Fixed

1. SF Bug 1582506 Docbook cannot handle <author>
2. SF Bug 1548189 Sections should not jump to <h4> for Accessibility reasons
3. SF Bug 1548180 Spaces dropped from index terms
4. SF Bug 1548154 XHTML index links should go to the topic
5. SF Bug 1545038 CommandLineInvoker is unfriendly towards spaces
6. SF Bug 1541055 topicref @id incorrectly uses NMTOKEN type
7. SF Bug 1530443 dost.jar relies on the incorrect behavior of Xerces
8. SF Bug 1473029 Syntax code makes overrides difficult
9. SF Bug 1470101 Metadata in topics is left out of XHTML headers
10. SF Bug 1470077 Choicetable headers create attribute inside attribute
11. SF Bug 1470057 Step template creates attributes after creating tags
12. SF Bug 1465947 <topichead> without children the whole branch to disappear
13. SF Bug 1465941 Keywords defined in map are ignored if <topicref> contains t
14. SF Bug 1465866 Problems in catalog-dita.txt
15. SF Bug 1460447 <morerows> not well supported in pdf tranformation.
16. SF Bug 1457187 'copy-to' doesn't actually copy files
17. SF Bug 1454835 OT renders files referenced via conref only
18. SF Bug 1427808 Should be easier to modify link attributes in XHTML
19. SF Bug 1422182 @colname renaming needs to apply to @namest and @nameend
20. SF Bug 1417820 fo and docbook outputs can\\'t handle deep topic dirs
21. SF Bug 1368997 PDF Vertical list of author redundancy

1 SourceForge Patch Added

1. SF Patch 1503296 Refactor of HTMLHelp inifiles creation

1 SourceForge RFE Added

1. SF RFE 1160960 Enh: Toolkit should work with both both *.dita and *.xml

DITA OT release 1.2.2

Release 1.2.2 is a maintenance release to fix defects and make patches based on release 1.2.1.

Improvements

1. Chinese support in WORD RTF
2. Improve plug-in architecture in plug-in dependency handling

SourceForge Changes

1. SF Bug 1461642 Relative paths in toolkit.
2. SF Bug 1463756 TROFF output is not usable
3. SF Bug 1459527 Properties elements should generate default headings
4. SF Bug 1457552 FO gen-toc does not work right for ditamaps and bookmaps
5. SF Bug 1430983 Specialized indexterm does not generate entries in index
6. SF Bug 1363055 Shortdesc disappears when optional body is removed
7. SF Bug 1368403 The dita2docbook transformation lacks support for args.xsl
8. SF Bug 1405184 Note template for XHTML should be easier to override
9. SF Bug 1407646 Map titles are not used in print outputs
10. SF Bug 1409960 No page numbers in PDF toc
11. SF Bug 1459790 Related Links omitted when map references file#topicid

12. SF Bug 1428015 Topicmerge.xsl should leave indentation alone
13. SF Bug 1429400 FO output should allow more external links
14. SF Bug 1405169 Space inside XHTML note title affects CSS presentation
15. SF Bug 1402377 Updated translations for Icelandic
16. SF Bug 1366845 XRefs do not generate page numbers
17. SF Patch 1326450 Make ${basedir} mine
18. SF Patch 1328264 FOP task userconfig file
19. SF Patch 1385636 Tweaks to docbook/topic2db.xsl
20. SF Patch 1435584 Recognize more image extensions
21. SF Patch 1444900 Add template for getting input file URI
22. SF Patch 1460419 Add a new parameter /cssroot:{args.cssroot}
23. SF Patch 1460441 map2hhp [FILES] include
24. SF RFE 1400140 Add a new parameter /cssroot:{args.cssroot}

DITA OT release 1.2.1

Release 1.2.1 is a maintenance release to fix defects and make patches based on release 1.2.

Improvements

1. Corrupt table generated in WORD RTF is fixed
2. Pictures are merged into the WORD RTF instead of creating links to them
3. lq element is supported in WORD RTF
4. Generated text can be translated to different languages in WORD RTF
5. In WORD RTF, if no <choptionhd> given, head will be generated in table

SourceForge Changes

1. SF Bug 1460451 Spaces preserving methods are different among tags.
2. SF Bug 1460449 Nested list can not be well supported.
3. SF Bug 1460445 h2d stylesheet cannot handle HTML files within namespace.
4. SF Bug 1431229 hardcoded path in MessageUtils.java
5. SF Bug 1408477 <desc> element is not handled inside xref for XHTML
6. SF Bug 1398867 ampersands in hrefs (on xref and link) cause build to fail
7. SF Bug 1326439 filtered-out indexterms leak into index through dita.list
8. SF Bug 1408487 Short description is not retrieved for <xref> element
9. SF Bug 1407454 XHTML processing for <alt> is incomplete
10. SF Bug 1405221 Some table frames ignored in dita->xhtml
11. SF Bug 1414398 Cannot set provider for Eclipse help transformation
12. SF RFE 1448712 add support for /plugins directory in plug-in architecture

DITA OT release 1.2

DITA open toolkit Release 1.2 is a major release to add new functions, fulfill new requirements, make some function enhancements and fix bugs over release 1.1.2.1.

Important Change

DITA-OT 1.2 offers new error handling and logging system. If you invoke your transformation by using java command line where new error handling and logging system is mandatory, you need to set the *CLASSPATH* Environment Variable for `dost.jar`. If you invoke your transformation by using an ant script, you need to do one more step after the setting above. That is adding a parameter in your command to invoke an ant script. For example, use `ant -f ant\sample_xhtml.xml -logger org.dita.dost.log.DITAOTBuildLogger` instead of `ant -f ant\sample_xhtml.xml` to start a transformation defined in ant script file ant\sample_xhtml.xml.

New Functions

1. **New plugin archiecture**

   DITA Open Toolkit 1.2 provides a new function to help users to download, install and use plug-ins and help developers create new plug-ins for DITA Open Toolkit.
2. **Transformation to wordrtf**

   DITA Open Toolkit 1.2 provides DITA to Word transforming function to transform DITA source files to output in Microsoft® Word RTF file.
3. **HTML to DITA migration tool**

   DITA Open Toolkit 1.2 provides a HTML to DITA migration tool, which migrates HTML files to DITA files. This migration tool originally comes from the developerWorks publication of Robert D. Anderson's how-to articles with the original h2d code.
4. **Problem determination and log analysis**

   In DITA Open Toolkit 1.2, a new logging method is supported to log messages both on the screen and into the log file. The messages on the screen present user with the status information, warning, error, and fatal error messages. The messages in the log file present user with more detailed information about the transformation process. By analyzing these messages, user can know what cause the problem and how to solve it.
5. **Open DITA User Guide for conditional processing**

   In DITA Open Toolkit 1.2, a new user guide which can help users to use conditional processing is added to toolkit document.
6. **Include the OASIS version langref**

   In DITA Open Toolkit 1.2, a new OASIS version of language reference for DITA standard is added to toolkit document.
7. **Document adapt to OASIS DITA 1.0.1 DTDs**

   DITA DTD files are updated to 1.0.1 version in DITA Open Toolkit 1.2.

Other Changes

1. SF Bug 1304545 Some folders were copied to DITA-OT's root directory
2. SF Bug 1328689 Stylesheet links in HTML emitted with local filesystem paths
3. SF Bug 1333481 Mapref function does not work for maps in another directory
4. SF Bug 1343963 Blank index.html generated for ditamap contains only reltabe
5. SF Bug 1344486 java.io.EOFException thrown out when reading ditaval file
6. SF Bug 1347669 Path Spec. in nested DITA maps
7. SF Bug 1357139 filtering behavior doesn't conform to spec
8. SF Bug 1358619 The property.temp file gets cleaned out by default
9. SF Bug 1366843 XRefs do not generate proper links in FO/PDF
10. SF Bug 1367636 dita2fo-elems.xsl has strange line breaks
11. SF RFE 1296133 Enable related-links in PDF output
12. SF RFE 1326377 Add a /dbg or /debug flag for diagnostic info
13. SF RFE 1331727 Toolkit need to run on JDK 1.5.x(only support to run under Sun JDK 1.5 with saxon in normal case)
14. SF RFE 1357054 Be more friendly towards relative directories
15. SF RFE 1357906 Provide a default output directory
16. SF RFE 1368073 Enable plugins for DITA open toolkit
17. SF RFE 1379518 Clearer error messages and improved exception handling
18. SF RFE 1379523 DITA to Rich Text Format (.rtf) file
19. SF RFE 1382482 plugin architecture of DITA-OT

DITA OT release 1.1.2.1

Release 1.1.2.1 is a full build to provide an urgent fix to fix the following critical problem which users found in release 1.1.2.

- SF Bug 1345600 The build process failed when run "Ant all" in release 1.1.2

For this fix, we have restored all the source DITA files in 'doc' and directories in the binary packages.

Note that the original parameter "args.eclipse.toc" in "Ant tasks and script" was separated to "args.eclipsehelp.toc" for DITA-to-Eclipse help transformation, and "args.eclipsecontent.toc" for DITA-to-dynamic Eclipse content transformation.

Another issue is that we found there is a mismatch in the document and the toolkit behavior when you are trying to use the following command

```
ant -f conductor.xml -propertyfile ${dita.temp.dir}/property.temp.
```

Now we have updated the documentation. Please refer to the topic 'Building DITA output with Java command line' on our website for more details.

These updates do not affect standard operation of the toolkit. The main goal of this minor release to enable new users of the toolkit to run the installation verification tests without failure.

DITA OT release 1.1.2

Release 1.1.2 is a maintenance release to fix defects and make patches based on release 1.1.1.

But there are certain limitations and unfixed bugs in this release, such as,

- Bug 1343963 Blank index.html generated for ditamap contains only reltabe
- Bug 1344486 java.io.EOFException thrown out when reading ditaval file

Please check the current 'open' bugs on the SourceForge bugs tracker.

Changes

1. SF Bug 1297355: Multilevel HTML Help popup shows filenames
2. SF Bug 1297657: Update for Supported Parameters page
3. SF Bug 1304859: Toolkit disallows repetition of topic ID within map
4. SF Bug 1306361: Fatal error in published ditamap example
5. SF Bug 1306363: common.css not compiled with htmlhelp
6. SF Bug 1311788: DTD references not resolved
7. SF Bug 1314081: Fix catalog entries in catalag-ant.xml for OASIS DTDs
8. SF Bug 1323435: wrong system id for html output used in validation
9. SF Bug 1323486: HTML Help subterm indexes not sorted
10. SF Bug 1325290: JavaHelp output does not work for Russian
11. SF Bug 1325277: File missing from the map causes abend
12. SF Patch 1253783: dita2fo-links relative hrefs
13. SF Patch 1324387: In xslfo, groupchoice var prints extra | delimiter
14. SF RFE 1324990: Installation Guide

Parameter Changes

1. The original parameter "args.eclipse.toc" in "Ant tasks and script" was separated to "args.eclipsehelp.toc" for dita2eclipsehelp transformation, and "args.eclipsecontent.toc" for dita2eclipsecontent transformation.
2. Several parameters were added to the java command line interface, including "/javahelptoc","/javahelpmap","/eclipsehelptoc","/eclipsecontenttoc","/xhtmltoc".

Other Changes

Change to the "doc" directory, except "doc\langref" directory:

1. The source dita files and the generated HTML, CHM, and PDF files were separated into separate downloads.
2. The source package contains the source dita files.
3. The binary package contains the generated HTML, CHM, and PDF files.

DITA OT release 1.1.1

Release 1.1.1 is a maintenance release to fix defects and make patches based on release 1.1.

For patch 1284023, we are changing the name of the jar lib file from dost1.0.jar back to dost.jar because we believe we need to keep the jar file name consistent through various releases.

Changes

1. SF Bug 1196409: HTMLHelp output does not reference CSS
2. SF Bug 1272687: extra "../" link generated by topicgroup
3. SF Bug 1273751: revision flag using unavailable pictures
4. SF Bug 1273816: Index generation doesn't cope with multilevel well
5. SF Bug 1281900: Unnecessary comment and href typo
6. SF Bug 1283600: unecessary space in document cause invalid parameter of Ant
7. SF Bug 1283644: multipul document($FILTERFILE,/) doesn't work (XALAN)
8. SF Patch 1251609: pretargets xsl directory needs to use ${dita.script.dir}
9. SF Patch 1252441: Files in temp directory not deleted before build
10. SF Patch 1253785: Inline images in dita2fo-elems
11. SF Patch 1284023: change the name of jar file and remove the version name

DITA OT release 1.1

Release 1.1 is a major release to add new functions, fulfill new requirements, make some function enhancements and fix bugs over release 1.0.2.

1. **Adaptation to the new OASIS DITA standard**

   Release 1.1 implements the new OASIS DITA 1.0 standard for DITA DTDs and Schemas.

   DTDs of the previous release locate in the directory **dtd/dita132** and schemas of the previous release locate in the directory **schema/dita132**.
2. **Transformation to troff**

   Release 1.1 supports new troff output. Troff output looks like Linux man page output.
3. **XML catalog support**

   An XML catalog, which can consist of several catalog entry files, is a logical structure that describes mapping information between public IDs and URLs of DTD files. A catalog entry file is an XML file that includes a group of catalog entries. If you want to know more about XML catalog, please refer*XML Catalog*.

   A catalog entry can be used to locate a unified resource identifier (URI) reference for a certain resource such as a DTD file. An external entity's public identifier is used for mapping to the URI reference. The URI of any system identifier can be ignored.
4. **Topicref referring to a nested topic**

   The href attribute of the topicref is extended to quote a nested topic in a dita file.

   For example, in previous releases, href attribute is set like: href = "xxx.dita"; in release 1.1, href attribute can be set like: href = "xxx.dita#abc.dita".
5. **Globalization support**

Release 1.1 supports over 20 popular languages within the content of dita files. And it also provides translation function for DITA keywords to over 20 languages. Currently this globalization support fully applies to Eclipse Help and XHTML transformations, and partially applies to other transformations.

6. **Accessibility support**

   Accessibility support is now partially applies to PDF and XHTML transformations.
7. **Eclipse Content Provider Support**

   Please refer to *Eclipse Content Provider* for detail information.
8. **Index information in output**

   The output of HTML Help and Java Help transformations contain index information now.
9. **Mapref function**

   Mapref refers to a special usage of the `<topicref>` element as a reference to another ditamap file. This allows you to manage the overall ditamap file more easily. A large ditamap file can thus be broken down into several ditamap files, making it easier for the user to manage the overall logical structure. On the other hand, this mechanism also increases the reusability of those ditamap files. If you want to know more about mapref, please refer *Mapref*.
10. **TOC generation for Eclipse Help transformation**

    TOC generation now supported in transformation to Eclipse Help. Eclipse.
11. **Helpset generation for Java Help transformation**

    Helpset generation now supported in transformation to Java Help.
12. **New parameters supported in Java commands**

    In Java commands: /indexshow, /outext, /copycss, /xsl, /tempdir.
13. **New parameters supported in Ant scripts**

    In Ant scripts: args.indexshow, args.outext, args.copycss, args.xsl, dita.temp.dir

Other Changes

1. SF bug 1220569: Add XML Schema processing to DITA-OT
2. SF bug 1220644: Prompted ant--image does not link for single topic to PDF
3. SF bug 1229058: Add schema validation loading file for processing
4. SF RFE 1176855: Ant must be run from toolkit directory
5. SF RFE 1183482: Copy pre-existing html to output dir
6. SF RFE 1183490: Provide argument to specify the location of temp dir
7. SF RFE 1201242: override capability

DITA OT release 1.0.2

Release 1.0.2 is a maintenance release to fix defects and adds some minor enhancements in release 1.0.1.

Changes

1. SF Bug 1181950: format attribute should be set to 'dita' for dita topic
2. SF RFE 1183487: Document the usage of footer property
3. SF RFE 1198847: command line interface support
4. SF RFE 1198850: architecture document update
5. SF RFE 1200410: need explanation for dita.list
6. SF RFE 1201175: XML catalog support
7. SF Patch 1176909: Add template for getting image URI

DITA OT release 1.0.1

Release 1.0.1 is a maintenance release to fix defects and adds some minor enhancements in release 1.0.

Changes

1. Committer: maplink.xsl doesn't generate related links for second level referred topic
2. Committer: avoid infinite loop of conref
3. SF Bug 1160964: Can't point above the directory which contains the map file
4. SF Bug 1163523: Broken XPath expression in mappull.xsl
5. SF Bug 1168974: useless DRAFT param in FO transformation
6. SF Bug 1173162: generate null internal link destination in fo transformation
7. SF Bug 1173164: Not correctly use document() in dita2fo-links.xsl
8. SF Bug 1173663: All base directories are DITA-OT 1.0
9. SF Patch 1163561: XLST match patterns test for element names
10. SF Patch 1165068: FO hyperlinks and FOP-generated PDF bookmarks
11. SF Patch 1174012: Modification to sequence.ditamap

DITA OT release 1.0

The initial release of the Open Sourced DITA Toolkit introduces major architectural changes from the previous, developerWorks version of the Toolkit.

New features

1. A new, Java-based processing architecture that supports single-threaded execution throughout.
2. Ant-based orchestration of the processing environment, from preprocessing to transformation to any required post-processing.
3. A pre-processor core that supports conditional processing and conref resolution.
4. Map-driven processing that generates links for transformed topics.
5. A new DITA to HTML transform that replaces the previous topic2html_Impl.xsl core transform. This new core is based on requirements for high-volume usage within IBM for the past several years.

Ant-driven processing means that you can integrate the DITA processing tools into a seamless pipeline within supportive environments such as Eclipse.

The DTDs and Schemas in this version are based on those in the previous dita132 package with bug fixes. The DITA OS Toolkit will later support the OASIS 1.0 specification in its public review form.

DITA history on developerWorks (pre-Open Source)

Versions of the toolkit prior to Open Source are in the developerWorks XML Zone at this address: *DITA Downloads* Change logs for those versions are within the Readme files in each distribution.

DITA 1.2 Specification Support

The latest version of the DITA Open Toolkit supports the DITA 1.2 specification. Initial support for this specification was added in version 1.5 of the toolkit; versions 1.5.1 and 1.5.2 contain minor modifications to keep up with the latest drafts. The specification itself was approved at approximately the same time as DITA-OT 1.5.2, which contains the final versions of the DTD and Schemas.

Earlier versions of the DITA Open Toolkit contained a subset of the specification material, including descriptions of each DITA element. This material was shipped in source, CHM and PDF format. This was possible in part because versions 1.0 and 1.1 of the DITA Specification contained two separate specification documents: one for the architectural specification, and one for the language specification.

In DITA 1.2, each of these has been considerably expanded, and the two have been combined into a single document. The overall document is much larger, and including the same set of material would double the size of the DITA-OT package. Rather than include that material in the package, we've provided the links below to the latest specification material.

Highlights of DITA 1.2 support in the toolkit include:

- Processing support for all new elements and attributes
- Link redirection and text replacement using keyref
- New processing-role attribute in maps to allow references to topics that will not produce output artifacts
- New conref extensions, including the ability to reference a range of elements, to push content into another topic, and to use keys for resolving a conref attribute.
- The ability to filter content with controlled values and taxonomies, using the new Subject Scheme Map
- Processing support for both default versions of task (original, limited task, and the general task with fewer constraints on element order)
- Acronym and abbreviation support with the new <abbreviated-form> element
- New link grouping abilities available with headers in relationship tables
- OASIS Subcommittee specializations from the learning and machine industry domains (note that the core toolkit contains only basic processing support for these, but can be extended to produce related artifacts such as SCORM modules)

To find detailed information about any of these features, see the specification documents at OASIS. The DITA Adoption Technical Committee has also produced several papers to describe individual new features. In general, the white papers are geared more towards DITA users and authors, while the specification is geared more towards tool implementors, though both may be useful for either audience. The DITA Adoption papers can be found from that TC's main web page.

Tested platforms and tools

See which tools and platforms have been used in testing the DITA processing system.

The DITA processing system has been tested against the following platforms and tools:

Platform or tool	Tested version
OS	• Windows XP • OS X 10.6 • SLES 10
XSLT processor	• Xalan-J 2.6 • Xalan-J 2.7 • Xalan-J 2.7.1 • Saxon 6.5 • Saxon 9 • Saxon-B 9.1 • Saxon-HE/EE 9.3 **Note:** XSLT 2.0 standard is not officially supported yet, due to the reliance by some users on Xalan.
JDK	• IBM 1.5 • IBM 1.6 • Oracle 1.5

Platform or tool	Tested version
	• Oracle 1.6
Ant	• Ant 1.7.1 • Ant 1.8.2

Using DITA transforms

The core transforms of the DITA Toolkit represent an implementation of all processing defined by OASIS in the DITA specification.

Pre-process

A pre-process is done before the main transformation. The input of the pre-process is dita files (maps and topics) and the output of the pre-process is often referred to as "normalized" dita files. The pre-process stage resolves several common DITA behaviors, such as resolving conref attributes, resolving keyref values, and adding links based on the hierarchy and relationship tables in a map. The normalized dita files are in a temporary directory. Most DITA transforms should use this common pre-process setup.

Available core transforms

A core DITA transform is the basic set of templates that support all the elements of a topic. This set is the basis for the following processing of any specialized element. Core transforms handle one topic instance, or nested set of topics, at a time. The DITA Toolkit provides several core transforms:

XHTML	DITA topic to XHTML page transform.
PDF	DITA topic or map to PDF, using XSL Formatting Objects
RTF	DITA topic or map to RTF; note that the output for RTF is more limited than for PDF, particularly for tables.
TROFF	DITA topic to TROFF; complex table markup is not supported by the TROFF transform

Available special output formats

Additional map-driven tools support transforming sets of topics into special output formats, including:

map2eclipsehelp (map2elipse.xsl)	This transform generates table of content for help contents in Eclipse.
Web page (map2htmtoc.xsl)	This transform generates a set of web pages with an index page that is ready to place on a Web site.
map2htmlhelp (map2hhc.xsl map2hhp.xsl)	This transform generates hhc and hhp file for the compilation of Html Help.
map2javahelp (map2JavaHelpToc.xsl map2JavaHelpMap.xsl)	This transform generates table of content and jhm file for Java Help.

Invoke the complete transformation

The complete transformation including pre-process can be executed by an ant script. There are some examples of simple ant scripts in the directory `samples/ant_sample`. See *Running Ant* on page 35 for more details about Ant processing.

Building DITA output with Ant

Ant is an open tool that the DITA Open Toolkit uses to manage builds.

Introduction to Ant

Ant is a Java-based, open source tool provided by the Apache Foundation to declare a sequence of build actions. It is well suited for development builds as well as document builds. The "Full Easy Install" version of the toolkit ships with a copy of Ant.

DITA provides a set of XSLT scripts for producing different types of documentation, such as help output in Eclipse, Java Help and HTML Help, XHTML pages, and PDF. The DITA-OT uses Ant to manage these scripts, as well as to manage additional intermediate steps written in Java.

Note: The following instructions and the associated *build.xml* and *ditatargets.xml* files are for the Java (and above), Ant , FOP , and Saxon-B releases. These instructions are likely to need some adjustment for other versions of these components and for specific environments.

Setting up Ant

This topic describes how to set up an Ant environment.

The *Java Development Kit (JDK)* and *XSLT processor* should be installed before setting up Ant.

Note: The "Full Easy Install" version of the DITA-OT comes with the latest tested versions of Ant, Saxon (a common XSLT processor), and other tools like FOP. If you are using the "Full Easy Install" package, you may run the "startcmd" batch file to set up an environment for all required libraries. When using this method, Java is the only tool that must be installed separately (it cannot be included in the DITA-OT due to licensing restrictions). For a full list of the tools that come in each DITA-OT package, see *DITA-OT Packages* at dita.xml.org.

1. Download and extract the Ant package file (available on *http://ant.apache.org/bindownload.cgi*) into a directory of your choice.
2. Set up environment variable.
   - On a Windows platform, follow these steps:
     - Set the JAVA_HOME environment variable, if it is not already defined: `set JAVA_HOME=<jdk_dir>`
     - Set the ANT_HOME environment variable: `set ANT_HOME=<ant_dir>`
     - Add the Ant binary directory to the PATH environment variable: `set PATH=%PATH%;<ant_dir>\bin`
   - On Linux, follow these steps:
     - Set JAVA_HOME as follows: `export JAVA_HOME=<jdk_dir>`
     - Set the ANT_HOME as follows: `export ANT_HOME=<ant_dir>`
     - Add Ant to the PATH as follows: `export PATH=$PATH:<ant_dir>\bin`
3. Optional: If you have installed FOP to generate PDF output, see *DITA installation* for additional information on setting it up.

Running Ant

After setting up the Ant environment, you can build the DITA output by running the `ant` command.

Here are some samples to explain how to use Ant to build sample output in the DITA directory.

Note: To run the Ant demo properly, you should switch to the **DITA installation directory** under the command prompt. If you are using the "Full Easy Install" package, running the "startcmd" batch file in that directory will give you a prompt that is already set up for the following commands.

- You can build all demos in the DITA directory

  Input `ant all -f build_demo.xml`

  The building process will create an **/out/** directory and put the output files in subdirectories that parallel the source directory.
- You can also rebuild specific DITA sample files after running the full demo.

  First you need to remove part of the output by specifying a "`clean`" target, and then rebuild the output. For example: To rebuild FAQ demo, run the following two commands:

  ```
  ant clean.demo.faq -f build_demo.xml
  ant demo.faq -f build_demo.xml
  ```

  Note: To find out the complete list of targets you can clean and build, check the *name* attributes for the target elements within the *build_demo.xml* file. Or, input `ant -projecthelp -f build_demo.xml` for a full list information.
- You can also build your own selections using a prompted build.

  Input `ant -f build_demo.xml`

  Ant will prompt you for the input file, output directory, and transform type. Values on these parameters are case sensitive.

To set up your own reusable Ant builds, with access to the full range of DITA-OT parameters, start by using the samples provided in `samples/ant_samples/`. That directory contains a sample Ant script for each common output format.

Note: To troubleshoot problems in setting up Java, Ant, Saxon, or FOP, you will get better information from the communities for those components rather than the communities for the DITA. Of course, if you find issues relevant to the DITA XSLT scripts (or have ideas for improving them), you are encouraged to engage the DITA community.

Ant tasks and scripts

This topic describes detailed Ant tasks and scripts.

The build process including pre-process can be called by using an Ant script. The most important Ant script files are called `build.xml`, `build_general.xml`, and `build_preprocess.xml`; together they define common pre-processing and output transformation routines, as well as extension points that allow DITA-OT Plug-ins to add to this common processing.

The following table lists many general parameters to the DITA-OT transforms.

Table 1: General Parameter Table

Parameter	Description	Required
basedir	The path of the working directory for transformations, it will be the base of relative paths specified by other parameters. **Note:** • If input is relative, it will be set relative to the current directory. • In Ant scripts, the default is that specified in the Ant buildfile. • In Java command line, the default is current directory.	No
dita.dir	The absolute path of the toolkit's home directory.	No

Parameter	Description	Required
args.input	The path and name of the input file. This argument should be in the same upper or lower case with the filename on file system. **Note:** This parameter must be provided if `dita.input` and `dita.input.dirname` not be provided.	No
dita.input	The name of the input file . **Note:** This parameter must be provided if `args.input` not be provided. And this parameter must be used together with the `dita.input.dirname` parameter. The result of this combination is equivalent to use only the `args.input` parameter. It is an alternative way to specify the path and name of the input file. ***DEPRECATED*** *- use* `args.input` *instead.*	No
dita.input.dirname	The input directory which contains the input file. **Note:** This parameter must be provided if `args.input` not be provided. And this parameter must be used together with the `dita.input` parameter. The result of this combination is equivalent to use only the `args.input` parameter. It is an alternative way to specify the path and name of the input file. ***DEPRECATED*** *- use* `args.input` *instead.*	No
dita.temp.dir	The directory of the temporary files. The default is 'temp'.	No
output.dir	The path of the output directory.	Yes
dita.extname	The file extension name of the input topic files, for example, '.xml' or '.dita'. The default is '.xml'.	No
args.xsl	The xsl file to replace the default xsl file. It will replace dita2docbook.xsl in docbook transformation, dita2fo-shell.xsl in pdf transformation, dita2xhtml.xsl in xhtml/eclipsehelp transformation, dita2rtfImpl.xsl in word transformation and dita2html.xsl in javahelp/ htmlhelp transformation.	No
dita.input.valfile	The name of the file containing *filter/flagging/revision* information.	No
args.draft	Default "hide draft & required-cleanup content" processing parameter (“no”= hide them); “no” and “yes” are valid values; non-“yes” is ignored.	No
args.artlbl	Default "output artwork filenames" processing parameter; “no”and “yes”are valid values; non-“yes” is ignored.	No
clean.temp	The parameter to specify whether to clean the temp directory before each build. Only "no" and "yes" are valid values. The default is yes.	No
args.logdir	The directory used to keep generated log files. Default will be output directory. **Note:** If several transforms running batchly, e.g., ant all: • If the user has specified a common logdir for all transformations, it will be used as log directory. • If the user hasn't specified a common dir for all transformations:	No

Parameter	Description	Required
	• If all transformations have same output directory, the common output direcory will be used as log directory. • If there is no same output directory for all transformations, the `basedir` will be used as default log directory.	
validate	The parameter to specify whether the ditamap/dita/xml files to be validated. Only "true" and "false" are valid values. The default is true. **Note:** It is not recommended to turn off the validation function , which will cause unexpected error during transformation.	No
outer.control	The parameter to specify how to respond to the overflowing dita/topic files. Only "fail", "warn" and "quiet" are valid values. The default is warn. **Note:** The detailed introduction: • *fail*: Fail quickly if files are going to be generated/copied outside of that directory • *warn*: Complete if files will be generated/copied outside, but log a warning • *quiet*: Quietly finish with only those files (no warning or error)	No
generate.copy.outer	The parameter to specify how to deal with the overflowing dita/topic files. Only "1", "2" and "3" are valid values. The default is 1. **Note:** The detailed introduction: • *1*: Only generate/copy files that fit within the designated output directory. • *2*: Generate/copy all files, even those that will end up outside of the output directory. • *3*: the old solution,adjust the input.dir according to the referenced files. (not default option any more but keep this as the option of backward compatibility).	No
onlytopic.in.map	The parameter to specify whether the referenced dita/topic files which are not referenced by ditamap files should be resolved. Only "true" and "false" are valid values. The default is false.	No

The following table lists general parameters that apply to the various HTML and XHTML based output formats.

Table 2: General Parameter Table for Tasks(dita2xhtml,dita2htmlhelp,dita2javahelp,dita2eclipsehelp)

Parameter	Description	Required
args.indexshow	The parameter to specify whether each index entry should display within the body of the text itself. Only "no" and "yes" are valid values.	No
args.copycss	The parameter to specify whether copy user specified css files to the directory specified by `{args.outdir}${args.csspath}`. "no" and "yes" are valid values. Default is "no".	No

Parameter	Description	Required
args.outext	The output file extension name for generated xhtml files. Typically, '.html' or '.htm' can be used as the extension name for the generated xhtml files. You can also specify other extension name. The default is '.html'.	No
args.css	User specified css file, it can be a local file or remote file from website. **Note:** If `${args.csspath}` is an URL, the `${args.css}` should be a filepath relative to the URL.	No
args.cssroot	The root directory of user specified css file. **Note:** If this parameter is set, the `${args.css}` should be a filepath relative to `args.cssroot`.	No
args.csspath	The path for css reference. Default is no path. **Note:** • If `${args.csspath}` is an URL like path, it should starts with `http://` or `https://`. For example: `http://www.ibm.com/css`. • Local absolute paths is not supported for `${args.csspath}`. • Use '`/`' as the path separator and don't append separator at last. For example: `css/mycss`.	No
args.hdf	The name of the file containing XHTML to be placed in the HEAD area.	No
args.hdr	The name of the file containing XHTML to be placed in the BODY running-heading area.	No
args.ftr	The name of the file containing XHTML to be placed in the BODY running-footing area.	No

Several additional parameters are available for other output formats.

XHTML (no navigation)

Table 3: Parameters for the dita2xhtml target

Parameter	Description	Required
args.xhtml.toc	The root file name of the output xhtml toc file in xhtml transformation. The default is 'index'.	No

Eclipse help output

Table 4: Parameters for the dita2eclipsehelp target

Parameter	Description	Required
args.eclipsehelp.toc	The root file name of the output eclipsehelp toc file in eclipsehelp transformation. The default is the name of input ditamap file.	No
args.eclipse.provider	The provider name of the eclipse help output. The default value is DITA.	No

Parameter	Description	Required
args.eclipse.version	The version number of the eclipse help output. The default value is 1.0	No

Eclipse Content (also known as normalized or dynamic DITA)

Table 5: Parameters for the dita2eclipsecontent target

Parameter	Description	Required
args.eclipsecontent.toc	The root file name of the output Eclipse content provider toc file in eclipsecontent transformation. The default is the name of input ditamap file.	No
args.eclipse.provider	The provider name of the eclipse help output. The default value is DITA.	No
args.eclipse.version	The version number of the eclipse help output. The default value is 1.0	No

HTML Help

Table 6: Parameters for the dita2htmlhelp target

Parameter	Description	Required
args.dita.locale	The locale used for sorting indexterms. If no locale specified, the first occurrence of "xml-lang" will be used as default locale; If no "xml-lang" found, "en-us" will be used by default.	No
args.htmlhelp.includefile	The parameter to specify the file that need to be included by the HTMLHelp output.	No

Java Help

Table 7: Parameters for the dita2javahelp target

Parameter	Description	Required
args.javahelp.toc	The root file name of the output javahelp toc file in javahelp transformation. The default is the name of input ditamap file.	No
args.javahelp.map	The root file name of the output javahelp map file in javahelp transformation. The default is the name of input ditamap file.	No
args.dita.locale	The locale used for sorting indexterms. If no locale specified, the first occurrence of "xml-lang" will be used as default locale; If no "xml-lang" found, "en-us" will be used by default.	No

RTF output: The transform from DITA to Word RTF does not support the `args.artlbl` parameter from the general parameters table.

Legacy PDF output: The default PDF output, which was initially developed as a plug-in to the toolkit, is not described here. The following parameters are only for the older PDF transform, now known as the "legacypdf" transform.

Table 8: Parameters to the LEGACY PDF TRANSFORM dita2legacypdf

Parameter	Description	Required
args.fo.img.ext	The extension name of image file in pdf transformation. Only '.jpg', '.gif' are valid value. The default is '.jpg'.	No

Parameter	Description	Required
	Note: Only one extension supported in the same transformation, image files with other extensions will be renamed to the specified extension.	
args.fo.output.rel.links	The parameter to specify whether output related links in pdf transformation. "yes" and "no" are valid values. Default is "no". **Note:** Any value that is not "yes" is regarded as "no".	No
args.fo.userconfig	The parameter to specify the user configuration file for FOP.	No

Sample ant script

These ant scripts are in `samples/ant_sample` directory. They are simple and easy to learn. From these files, you can learn how to write your own Ant script to build your own process.

Here is a sample template for writing an Ant script that executes transformation to xhtml in `samples/ant_samples` directory:

```
<?xml version="1.0" encoding="UTF-8" ?>
<project name="@PROJECT.NAME@_xhtml" default="@DELIVERABLE.NAME@2xhtml"
 basedir=".">

  <property name="dita.dir" value="${basedir}${file.separator}..
${file.separator}.."/>

  <import file="${dita.dir}${file.separator}integrator.xml"/>

  <target name="@DELIVERABLE.NAME@2xhtml" depends="integrate">
    <ant antfile="${dita.dir}${file.separator}build.xml" target="init">
      <!-- please refer to the toolkit's document for supported parameters,
 and
           specify them base on your needs -->
      <property name="args.input" value="@DITA.INPUT@"/>
      <property name="output.dir" value="@OUTPUT.DIR@"/>
      <property name="transtype" value="xhtml"/>
    </ant>
  </target>

</project>
```

To use this template, modify the following items:

- Replace `@PROJECT.NAME@` with the name of your project, such as "MyDocs".
- Replace `@DELIVERABLE.NAME@` with the name of your deliverable, such as "installDocs".
- Replace `@DITA.INPUT@` with the name of your input file (using either a full path or a relative path from the location of this template).
- Replace `@OUTPUT.DIR@` with the desired output directory (using either a full path or a relative path from the location of this template).

Once you have updated these items, you can run your build with the following command:

```
ant -f samples/ant_sample/template_xhtml.xml
```

The build will convert your input file to XHTML. Note that the build directly calls the Ant script "build.xml", which is a common entry point for DITA-OT builds; it in turn imports all of the scripts mentioned above.

Building DITA output with Java command line

The DITA Open Toolkit provides a Java command line interface as an alternative for users with little knowledge of Ant. Most parameters available to the Ant builds are also available using the Java command line.

Note: This information is currently being integrated with the parameter information in the Ant Quick Start Guide. Newer information on Java and Ant parameters can be found in that document here: *Ant Properties for DITA-OT*

Note: The Java command line interface is simply a wrapper around the Ant interface; it takes the simplified parameters as input, converts them to Ant parameters, and then runs an Ant build. This means that in general, applications embedding the toolkit are better off simply calling Ant directly. For individual builds, the additional Java overhead is minimal, but for repeated or server based builds, it the extra memory usage may become more of an issue.

Running a Java based build.

If you are using the "Full Easy Install" package, running the startcmd batch file will set up a build environment for you and put you in the correct directory. If you are not using this method, you must set up all of your tools (Ant, XSLT, FOP, etc) before running the build.

1. Change into the DITA Open Toolkit installation directory.
2. On the command line, enter the following command:

```
java -jar lib/dost.jar /i:samples/sequence.ditamap /outdir:out /
transtype:xhtml
```

This particular example creates a properties file, and then calls Ant using this properties to build the sample `sequence.ditamap` file to XHTML. The output is placed in the out/ directory. You can add other parameters to this properties file and then run the build directly from Ant; see *Table 9: Table of supported parameters* on page 42 for details on additional parameters.

Note:

1. In this example, the character slash preceded by a space is the separator for each parameter.
2. Currently, the parameters `/filter`, `/ftr`, `/hdr`, and `/hdf` require an absolute path.
3. The properties file is saved in the `${args.logdir}` directory. The following command provides an example using this properties file:

```
ant -f conductor.xml -propertyfile ${args.logdir}/property.temp
```

4. To see a list of all supported parameters from the Java command line, run the following command with no additional parameters:

```
java -jar lib/dost.jar
```

Supported parameters

The following table lists many supported parameters that you can set with this tool. The equivalent Ant parameter names are specified within braces. To get a full list of parameters, run the following command with no additional options: `ava -jar lib/dost.jar`

Table 9: Table of supported parameters

Parameter	Description
/basedir:{basedir}	The path of the working directory for transformations, it will be the base of relative paths specified by other parameters.

Parameter	Description
	Note: • If input is relative, it will be set relative to the current directory. • In Ant scripts, the default is that specified in the Ant buildfile. • In Java command line, the default is current directory.
/ditadir:{dita.dir}	The absolute path of the toolkit's home directory.
/i:{args.input}	The path and name of the input file. This argument should be in the same upper or lower case with the filename on file system. **Note:** This parameter must be provided if `dita.input` and `dita.input.dirname` not be provided.
/if:{dita.input}	The name of the input file . **Note:** This parameter must be provided if `args.input` not be provided. And this parameter must be used together with the `dita.input.dirname` parameter. The result of this combination is equivalent to use only the `args.input` parameter. It is an alternative way to specify the path and name of the input file. ***DEPRECATED*** *- use* `args.input` *instead.*
/id:{dita.input.dirname}	The input directory which contains the input file. **Note:** This parameter must be provided if `args.input` not be provided. And this parameter must be used together with the `dita.input` parameter. The result of this combination is equivalent to use only the `args.input` parameter. It is an alternative way to specify the path and name of the input file. ***DEPRECATED*** *- use* `args.input` *instead.*
/outdir:{output.dir}	The path of the output directory.
/tempdir:{dita.temp.dir}	The directory of the temporary files. The default is 'temp'.
/ditaext:{dita.extname}	The file extension name of the input topic files, for example, '.xml' or '.dita'. The default is '.xml'.
/transtype:{transtype}	The transformation type. Currently, the supported values include xhtml, pdf, javahelp, eclipsehelp, htmlhelp, eclipsecontent, troff, wordrtf and docbook.

Parameter	**Description**
/filter:{dita.input.valfile}	The name of the file containing *filter/flagging/ revision* information.
/draft:{args.draft}	Default "hide draft & required-cleanup content" processing parameter (“no”= hide them); “no” and “yes” are valid values; non-“yes” is ignored.
/artlbl:{args.artlbl}	Default "output artwork filenames" processing parameter; “no”and “yes”are valid values; non-“yes” is ignored.
/ftr:{args.ftr}	The name of the file containing XHTML to be placed in the BODY running-footing area.
/hdr:{args.hdr}	The name of the file containing XHTML to be placed in the BODY running-heading area.
/hdf:{args.hdf}	The name of the file containing XHTML to be placed in the HEAD area.
/csspath:{args.csspath}	The path for css reference. Default is no path. **Note:** • If `${args.csspath}` is an URL like path, it should starts with `http://` or `https://`. For example: `http://www.ibm.com/css`. • Local absolute paths is not supported for `${args.csspath}`. • Use '`/`' as the path separator and don't append separator at last. For example: `css/mycss`.
/css:{args.css}	User specified css file, it can be a local file or remote file from website. **Note:** If `${args.csspath}` is an URL, the `${args.css}` should be a filepath relative to the URL.
/cssroot:{args.cssroot}	The root directory of user specified css file. **Note:** If this parameter is set, the `${args.css}` should be a filepath relative to `args.cssroot`.
/copycss:{args.copycss}	The parameter to specify whether copy user specified css files to the directory specified by `{args.outdir}${args.csspath}`. "no" and "yes" are valid values. Default is "no".
/indexshow:{args.indexshow}	The parameter to specify whether each index entry should display within the body of the text itself. Only "no" and "yes" are valid values.
/outext:{args.outext}	The output file extension name for generated xhtml files. Typically, '.html' or '.htm' can be used as the extension name for the generated xhtml

Parameter	Description
	files. You can also specify other extension name. The default is '.html'.
/xsl:{args.xsl}	The xsl file to replace the default xsl file. It will replace dita2docbook.xsl in docbook transformation, dita2fo-shell.xsl in pdf transformation, dita2xhtml.xsl in xhtml/ eclipsehelp transformation, dita2rtfImpl.xsl in word transformation and dita2html.xsl in javahelp/ htmlhelp transformation.
/cleantemp:{clean.temp}	The parameter to specify whether to clean the temp directory before each build. Only "no" and "yes" are valid values. The default is yes.
/foimgext:{args.fo.img.ext}	The extension name of image file in pdf transformation. Only '.jpg', '.gif' are valid value. The default is '.jpg'. **Note:** Only one extension supported in the same transformation, image files with other extensions will be renamed to the specified extension.
/javahelptoc:{args.javahelp.toc}	The root file name of the output javahelp toc file in javahelp transformation. The default is the name of input ditamap file.
/javahelpmap:{args.javahelp.map}	The root file name of the output javahelp map file in javahelp transformation. The default is the name of input ditamap file.
/eclipsehelptoc:{args.eclipsehelp.toc}	The root file name of the output eclipsehelp toc file in eclipsehelp transformation. The default is the name of input ditamap file.
/eclipsecontenttoc:{args.eclipsecontent.toc}	The root file name of the output Eclipse content provider toc file in eclipsecontent transformation. The default is the name of input ditamap file.
/provider:{args.eclipse.provider}	The provider name of the eclipse help output. The default value is DITA.
/version:{args.eclipse.version}	The version number of the eclipse help output. The default value is 1.0
/xhtmltoc:{args.xhtml.toc}	The root file name of the output xhtml toc file in xhtml transformation. The default is 'index'.
/logdir:{args.logdir}	The directory used to keep generated log files. Default will be output directory. **Note:** If several transforms running batchly, e.g., ant all: • If the user has specified a common logdir for all transformations, it will be used as log directory. • If the user hasn't specified a common dir for all transformations:

Parameter	Description
	• If all transformations have same output directory, the common output direcory will be used as log directory. • If there is no same output directory for all transformations, the `basedir` will be used as default log directory.
/ditalocale:{args.dita.locale}	The locale used for sorting indexterms. If no locale specified, the first occurrence of "xml-lang" will be used as default locale; If no "xml-lang" found, "en-us" will be used by default.
/fooutputrellinks:{args.fo.output.rel.links}	The parameter to specify whether output related links in pdf transformation. "yes" and "no" are valid values. Default is "no". **Note:** Any value that is not "yes" is regarded as "no".
/fouserconfig:{args.fo.userconfig}	The parameter to specify the user configuration file for FOP.
/htmlhelpincludefile:{args.htmlhelp.includefile}	The parameter to specify the file that need to be included by the HTMLHelp output.
/validate:{validate}	The parameter to specify whether the ditamap/dita/xml files to be validated. Only "true" and "false" are valid values. The default is true. **Note:** It is not recommended to turn off the validation function , which will cause unexpected error during transformation.
/outercontrol:{outer.control}	The parameter to specify how to respond to the overflowing dita/topic files. Only "fail", "warn" and "quiet" are valid values. The default is warn. **Note:** The detailed introduction: • *fail*: Fail quickly if files are going to be generated/copied outside of that directory • *warn*: Complete if files will be generated/copied outside, but log a warning • *quiet*: Quietly finish with only those files (no warning or error)
/generateouter:{generate.copy.outer}	The parameter to specify how to deal with the overflowing dita/topic files. Only "1", "2" and "3" are valid values. The default is 1. **Note:** The detailed introduction: • *1*: Only generate/copy files that fit within the designated output directory.

Parameter	Description
	• *2*: Generate/copy all files, even those that will end up outside of the output directory. • *3*: the old solution,adjust the input.dir according to the referenced files. (not default option any more but keep this as the option of backward compatibility).
/onlytopicinmap:{onlytopic.in.map}	The parameter to specify whether the referenced dita/topic files which are not referenced by ditamap files should be resolved. Only "true" and "false" are valid values. The default is false.

Enhanced command line help

You can find the version of toolkit and the usage of the command line from the command line help by using the following commands:

```
java -jar lib/dost.jar -version

java -jar lib/dost.jar -h
```

You can see the brief description of the supported parameters in the command line window when you type a specific command. For example, `java -jar lib/dost.jar -h`

Problem determination and log analysis

Introduction

In the DITA Open Toolkit 1.2 or above, a new logging method is supported to log messages both on the screen and into the log file. The messages on the screen present user with the status information, warning, error, and fatal error messages. The messages in the log file present user with more detailed information about the transformation process. By analyzing these messages, user can know what cause the problem and how to solve it.

The logging method is based on Ant's Logger & Listener interface. By default, this logging method is disabled, and all the messages occur on the screen just like previous releases.

To start this new logging method, you need to follow the usage below:

- In Ant command, specify the logger by appending `-logger org.dita.dost.log.DITAOTBuildLogger` in the command parameters, for example:

  ```
  ant sample.web -logger org.dita.dost.log.DITAOTBuildLogger
  ```
- In Java command, the logger is specified internally, so you do not need to specify it again.

Analyze messages on the screen

During the building process, some information or messages occur on the screen to tell you about the status, warnings, errors, or fatal errors. You need to analyze the messages to solve the problems.

- If the build succeeded with some warning messages on the screen, it means that there are something incorrect within the user input parameters or source DITA files; but you can still get the correct output.

- If the build succeeded with some error messages on the screen, it means that there are something incorrect within the user input parameters or source DITA files; the output maybe not correct.
- If the build failed with fatal error message on the screen, it means that there are something illegal or invalid within the user input parameters or source DITA files; you may get no output, or wrong output.

Analyze messages in the log file

A log file in plain text format is generated in the log directory, which has a name combined with both input file name and transform type. You can open it and find more detailed information, which are helpful for solving problems. You can use the same way introduced above to analyze the messages and solve the problems.

The log directory can be specified by using the parameter `/logdir:{args.logdir}` for the output options.

Note: In some cases, there would be no log file generated:

- You have entered an invalid Ant command or Java command to start the toolkit.
- The log file with the same name in the same directory exists and can not be deleted.

Turn on debug mode

Debug mode is supported along with the new logging method. Under debug mode, diagnostic information, such as: environment variables, stack trace, will be logged into the log file. These information can help the user or developer to go deep into the problems and find the root cause.

By default, the debug mode is disabled. To turn on the debug mode, you need to follow the usage below:

- Append `-d` or `-debug` in Ant command.
- Append `/d` or `/debug` in Java command.

About message file

The message file is used to store the detailed log messages, these messages are read dynamically from this file. To ensure those messages can be read correctly during the transform process, the message file should be located properly. In some situations, the toolkit may fails to load the message file due to some exceptions thrown. Please refer to *Troubleshooting* on page 48 for detailed information.

For high level users and developers, there is a property `args.message.file` in the toolkit's ant script, it is used to config the message file, you can override it in your ant script.

Note: Due to the difference of underly implementation between Java, And, and XSL, the property `args.message.file` is only useful for Java and Ant; To keep the normal function of log handling, you still need to ensure there are files 'resource/messages.xml' and 'resource/messages.dtd' both in the toolkit's root directory and in the directory that you run the toolkit.

Troubleshooting

This section is used for identifying problems when installing and executing the DITA Open Toolkit.

1. Out of Memory Error

In some cases, you might receive a message stating the build has failed due to an "Out of Memory" error. In many cases this can be solved by switching from Xalan to Saxon as the default processor. Recent versions of the "Full Easy Install" toolkit distribution ship with Saxon instead of Xalan.

If that does not work, please follow the steps below to fix this problem:

1. For Windows, type `set ANT_OPTS=%ANT_OPTS% -Xmx256M` in the command prompt before running a build. Alternatively, you can add the value `-Xmx256M` to the `ANT_OPTS` environment variable.

   For Linux, type `export ANT_OPTS=${ANT_OPTS} -Xmx256M` in the command prompt before running a build.

2. Run the transformation again.

2. java.io.IOException: Can't store Document

In some cases, when you run the JavaHelp transformation, you might receive the exception above. This problem is caused by some HTML files unrelated with the current JavaHelp transformation were found under the output directory. Please follow the steps below to fix this problem:

1. Change into the output directory.
2. Clean the output directory.
3. Run the JavaHelp transformation again.

3. Failed to load message file

In some situations, the toolkit may fails to load the message file "messages.xml" and begin to throw exceptions.

To fix this problem, you need to check if the files 'resource/messages.xml' and 'resource/messages.dtd' exist in the toolkit. If not, please copy them from the toolkit's root directory.

4. Spaces in file names

Spaces in file names will cause trouble during the processing because Ant use space as the delimiter when processing batch files in a list. Please prevent using spaces in the name of dita files.

5. Stack Overflow

Sometimes, you will receive an error during the transformation which says the stack memory overflows. Please follow the steps below to fix the problem:

1. For Windows, type `set ANT_OPTS=%ANT_OPTS% -Xms512M` in the command prompt, you can also choose to add a new opition `-Xms512M` to the `ANT_OPTS` environment variable.

   For Linux, type `export ANT_OPTS=${ANT_OPTS} -Xms512M` in the command prompt.
2. Run the transformation again.

Known Limitations

Below are some known limitations categorized by module within the current release of the DITA Open Toolkit.

Transformation to PDF and Word RTF

1. You can change the styles of the output file by using tools in Microsoft® Word rather than specifying the styles before transforming.
2. If there is a cross reference referring to an URL in the DITA source file, the link should be completed defined with the proper internet protocol. For example, specify `http://www.ibm.com` instead of `www.ibm.com`.
3. Flagging, revision bar and filtering are not supported in PDF and Word RTF output.
4. The morerows attribute of the table element used to generate vertically merged cell is not supported in PDF output.
5. Style attributes for table are not supported in Word RTF output.
6. Complex cases dealing with tables in lists are not supported in Word RTF.
7. There may be no output style applied on contents of some tags in Word RTF output compared with other output.

HTML to DITA migration

1. Since Xalan doesn't allow to set the public and system IDs dynamically using a variable, when Xalan is used as the default XSLT processor, the output will contain:

```
<!DOCTYPE topic PUBLIC "{$publicid}" "{$systemid}">
```

Suggest to use Saxon as the processor to fix this problem. For other information on this problem, see the section "Other general migration notes" in the first developerWorks article.

2. Currently, the stylesheet can't handle HTML files within namespace like below:

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

Note: This limitation has been fixed in release 1.2.1, please refer to the *Migrating HTML to DITA* on page 50 for detail information.

DITA to Word output transform

The DITA Open Toolkit release 1.2 or above provides a DITA to Word transforming function to transform DITA source files to output in Microsoft® Word RTF file. The whole structure of the output file is the same as the structure designed in the ditamap file of the DITA source files.

To avoid losing files in the final output, make sure the ditamap file contains all topics which are cross referenced from within any individual topics.

Transforming DITA to Word output

You can use an ant script to transform DITA source files to word output. See *Transforming DITA to Word with Ant script* for more details.

You can also use the Java™ command for transform. See *Transforming DITA to Word with Java command* for more details.

Limitations

There are several limitations of the current DITA to Word transforming tool when processing. See *Known Limitations* on page 49 for details.

Migrating HTML to DITA

The DITA Open Toolkit release 1.2 or above provides a HTML to DITA migration tool, which migrates HTML files to DITA files. This migration tool originally comes from the developerWorks publication of Robert D. Anderson's how-to articles with the original h2d code. This migration tool is under "demo\h2d" directory. You can use it separately because it is not integrated into the main transformation of toolkit. The version in the toolkit is more recent, but the articles should be referenced for information on details of the program, as well as for information on how to extend it. There are links to the articles at the bottom of this page.

Preconditions

The preconditions to be considered before using the migration tool are listed below:

- The HTML file content must be divided among concepts, tasks, and reference articles. If not, the HTML files should be reworked before migrating.
- This migration tool is intended for topics. The HTML page should contain a single section without any nested sections.
- DITA architecture is focused on topics, information that is written for books needs to be redesigned in order to fit into a topic-based archiecture.
- This migration utility only works with valid XHTML files, HTML files must be cleaned up using HTML Tidy or other utility before processing.

Running examples

You can use the Ant script to migrate only one HTML file or all the HTML files in same directory each time. See *Migrating HTML to DITA with Ant script* on page 51 for more details.

You can also use the Java command for migration. See *Migrating HTML to DITA with Java command* on page 52 for more details.

Post conditions

There are also some post conditions to consider after processing:

- In some case, the tool cannot determine the correct way to migrate, it places the contents in a <required-cleanup> element, you should fix such elements in the output DITA files.
- Check the output DITA files. Compare them with the source HTML files and check if both contents are equivalent.

Known limitations

There are some known limitations within the current release, please refer to *Known Limitations* on page 49 for detailed information.

Extension points

The HTML2DITA migration tool helps extension in the following listed ways:

- The `genidattridbute` template can be overridden to change the method for creating the topic ID.
- The `gentitlealts` template can be overridden to change the ways of title generation.
- Override respond section in the tool to preserve the semantic of source, in case if the <div> or <span> element is used in regular structures.
- You can also migrate to another specialized DTD by overriding the original template base on the specific DTD and your required output.

Additional information

You can find the here original developerWorks publication via links below:

- *Migrating HTML to DITA, Part 1: Simple steps to move from HTML to DITA*
- *Migrating HTML to DITA, Part 2: Extend the migration for more robust results*

Migrating HTML to DITA with Ant script

Running example

1. Start the command window.
2. Navigate to the directory of the migration tool.
3. Use ant script to run the migration, on the command line, enter the following command:
```
ant -Dargs.input={file|direcotry} -Dargs.output={direcotry} -
Dargs.infotype={topic|concept|task|reference}
```

Note: The namespace problem listed in *Known Limitations* on page 49 has been fixed by adding a new preprocess step in the script in release 1.2.1.

You can also add other parameters to the command. See the following table for details.

Supported Parameters

The following table lists the supported parameters that you can set with the ant script.

Table 10: Table of supported parameters

Parameter	Descriptoin	Required
args.input	The input of the migration. It can be a file or directory. Default is current directory.	No.
args.include.subdirs	The parameter to specify if sub directories under the input directory is included. "yes" and "no" are valid values. Default is "no". **Note:** Any value that is not "yes" is regarded as "no".	No.
args.output	The output directory of genrated DITA files. Default is the current directory.	No.
args.infotype	The infotype of generated DITA files, topics, concept, task, and reference are valid values. Default is topic.	No.
args.dita.ext	The extension of generated DITA files. This extension also used to convert links that go to other DITA topic. ".dita" and ".xml" are valid values. Default is ".dita".	No.
args.xsl	The xsl file to replace the default xsl file.	No.
args.lang	The default language of output DITA files. Default is "en-us". **Note:** For supported language, please refer to `strings.xml` under the directory `${ditaot_dir}/xsl/common`.	No.

Migrating HTML to DITA with Java command

Running example

1. Start the command window.
2. Navigate to the directory of the migration tool.
3. (Optional) If the input HTML file contains namespace, you can remove it by hand, or running the command below:

   enter the following command when using Saxon:

   ```
   java com.icl.saxon.StyleSheet mytask.htm preprocess.xsl > mytask.htm
   ```

enter the following command when using Xalan:

```
java org.apache.xalan.xslt.Process -in mytask.htm -xsl preprocess.xsl -out
 mytask.htm
```

4. Use Saxon or Xalan directory to run the migration, on the command line,

   enter the following command when using Saxon:

   ```
   java com.icl.saxon.StyleSheet mytask.htm h2d.xsl infotype=task > mytask.dita
   ```

   enter the following command when using Xalan:

   ```
   java org.apache.xalan.xslt.Process -in mytask.htm -xsl h2d.xsl -out
    mytask.dita -param infortype task
   ```

You can also add other parameters to this properties file. See the following table for details.

Note: The output directory of the generated DITA file should exist, since the XSLT processor can't create it automatically.

Supported parameters

The following table lists the supported parameters that you can set with the java command.

Table 11: Table of supported parameters

Parameter	Description	Required
infotype	The infotype of generated DITA files. Topic, concept, task, and reference are valid values. Default is topic.	No.
dita-extension	The extension for links that go to other DITA topics. ".dita" and ".xml" are valid values. Default is ".dita". **Note:** The extension of the generated DITA file can't specified by this parameter, it only can be specified along with the output filename.	No.
FILENAME	It is used to determine the main topic's ID. **Note:** The FILENAME should ends with '.htm' or '.html'. Invalid ID characters, including all numbers, will be replaced with letters.	No.
default-lang	The default language of output DITA files. Default is "en-us". **Note:** For supported language, please refer	No.

Parameter	Description	Required
	to `strings.xml` under the directory `${ditaot_dir}/xsl/common`.	

Development Reference

Supporting two file extensions in one DITA map

DITA Open Toolkit supports two different file extensions, ".dita" and ".xml". Previous releases of DITA Open Toolkit do not support the transformation of DITA maps containing inconsistent file types, such as one DITA map containing both .dita and .xml files. Though you can create either .dita or .xml files, you cannot include both kinds of files in one DITA map. This makes file reuse difficult, because you have to change the file extensions manually make them consistent in one DITA map.

In DITA OT 1.3, you can include both .xml and .dita as the file extensions in one DITA map and transform the DITA map into your desired output without manually changing the file extensions.

If you include both .xml and .dita files in one DITA map, and specify /ditaext:.dita in Java command, the .xml files are transferred to .dita files and put in the temp directory together with the .dita files. If you specify /ditaext:.xml in Java command, all the .dita files are transferred to .xml files under the temp directory. The default process option is changing all files into .xml files.

Note:

- It is not suggested that you include files with the same root name but different extensions in the same directory because this might cause unexpected problems.
- Error messages and warning messages in the console might not reflect the real extension. For example, if there is an incorrect usage in a.dita, the warning message in the console might refer to a.xml, because a.dita was changed into a.xml in the temp directory.

You might use other file extensions together with .dita and .xml in one ditamap as well, such as .dit, but they are not tested in DITA OT 1.3 and thus you might take the risk of transformation failure.

Standard XML catalog resolver

In the previous releases of DITA Open Toolkit, a simple XML catalog resolver is enabled. You do not need to update the reference to dtd in DITA files when the file paths are changed; however, this simple implementation cannot be redistributed because it does not support standard XML catalogs.

In DITA OT 1.3, a standard XML catalog resolver is enabled so that the reference to dtd in DITA files does not need to be updated each time when you change the file paths on your workstation or use another workstation.

With this enhanced feature, when a developer makes a new specialization, the developer only needs to update the mapping between the new dtd file's system id (location relative to the catalog file) and public id (the id assigned by the developer in the head of the DITA or xml file which identifies the corresponding dtd file) in the catalog file (catalog-dita_template.xml), for example, `<public publicId="-//IBM//DTD DITA ABC//EN" uri="dtd/abc.dtd"></public>`.

This enhanced feature does not change the normal behavior of the toolkit.

Topic merge

The topic merge feature improves the build speed of DITA files and reduces the possibility of meeting the out of memory exception in the build process. As illustrated in the following figure, when you run the build in previous releases of DITA Open Toolkit, the build speed is slow and you are likely to get out of memory exception.

With this enhanced topic merge feature, you will be less likely to meet the out of memory exception error when you build output through DITA files. The intermediate merged file will keep the structure information in the DITA map, and the structured toc will be reflected in the output.

To know more about this topic feature, you can write a script file first. DITA OT 1.3 offers a module, `TopicMerge`, that helps you implement this feature. You can use this module to generate the merged files. A sample usage of this module is as follows.

sample.xml:

```
<project name="sample">
 <property name="dita.dir" value="${basedir}"/>
 <import file="${dita.dir}${file.separator}build.xml"/>

 <target name="premerge">
    <antcall target="preprocess">
    <param name="args.input" value="${input}"/>
  <param name="output.dir" value="${dita.dir}${file.separator}output"/>
  </antcall>
 </target>
 <target name="merge"  description="Merge topics" depends="premerge">
  <basename property="temp.base" file="${input}" suffix=".ditamap"/>
  <property name="temp.input"
 value="${basedir}${file.separator}${dita.temp.dir}${file.separator}${temp.base}"/
>
  <dirname property="temp.dir" file="${temp.input}"/>
  <pipeline message="topicmerge" module="TopicMerge"
   inputmap="${temp.dir}${file.separator}${temp.base}.ditamap"
   extparam="output=${dita.dir}${file.separator}output
${file.separator}${temp.base}_merged.xml;
      style=${dita.dir}${file.separator}xsl${file.separator}pretty.xsl" />
 </target>
</project>
```

Then, you need to type `ant -f sample.xml merge -Dinput="C:\DITA-OT1.3\test.ditamap`" in the command window.

Note: The path for -Dinput must be an absolute path

Working with documentation plug-in

You can use a template to develop documentation plug-in with DITA in Eclipse PDE and use DITA OT 1.3 to build and pack the final plug-in. When you want to develop a documentation plug-in with DITA in Eclipse, you cannot use the previous releases of DITA OT in Eclipse to transform DITA to HTML. Though previous releases of DITA OT support the feature to transform DITA files to Eclipse documentation plug-in, they are not integrated with Eclipse. With DITA OT 1.3 integrated with WPT, you can develop document plug-ins with DITA in Eclipse PDE and build and pack the final plug-in by taking the following steps.

1. Create a new PDE project in Eclipse, and apply the DITA template to the project by following the wizard.
2. Set the source directory, the main ditamap file, the output directory (default value is root directory of project), css storage directory (used to contain common.css, commonltr.css, and commonrtl.css), user customized .css file name, and conditional processing ditaval file in the wizard. **Use root as output directory** is selected as the default.

   You can also clear **Use root as output directory** and specify another output directory.

3. Create DITA files in the source directory and a ditamap to include the topic files that you created.
4. Optional: Import the DITA files into the `src` directory of the DITA plug-in project you just created.
   a) Right-click a directory that you want to put the imported files and select Import, and then File system.

   b) Select the directory under which you put the DITA files.

   c) Click Finish after you selected the DITA files under the specified directory. The DITA files are then imported to your DITA project.
5. Right click build.xml, select Run As, and then ANT Build.

   Note: If you're using SUN JDK, please download and use the latest Xalan. The Xalan shipped with SUN JDK has some issue that will cause the build failure. You can use the latest Xalan by selecting **ANT Build ...** and include the all of Xalan's jar files in Classpath.

   After the transformation, the output is in the output directory set in build.xml. Refresh the project after the build is successful.

6. Edit the plug-in description of the property file MANITEST.MF in the plug-in editor after you run the ANT build successfully.
   a) Click MANITEST.MF to go to the Overview page.

   b) Edit Dependencies to include org.eclipse.help.

   c) Edit Extensions to add org.eclipse.help.toc; right click the added prgeclipse.help.toc, and select New, and then toc.

   d) Edit the Build Configuration to include the out directory or the directory you specified in *Step 2*.

   e) Save the changes you made to the property file MANITEST.MF.
7. Export the output to a documentation plug-in.

   Note: build.xml can be customized to meet the requirement of headless build.

   a) Select **File** > **Export** ; select Deployable plug-ins and fragments and click **Next**.

   b) Select the plug-in you want to export and specify a directory under which you want to put the plug-in package.

   c) Click **Finish** to export the plug-in package.