

Antenna House PDF5 plug-in

# User's Guide

---

For DITA Open Toolkit

Antenna House, Inc. Japan 21 4 2015  
PDF5-001 1  
2009 2015 Antenna House, Inc.

---

# Preface

## • About Antenna House I18n Index Library

Originally this PDF5 plug-in was created for Antenna House I18n Index Library. I18n Index Library is a Java library that makes index pages in the various languages used by the DocBook, DITA to XSL-FO stylesheet.

However this plug-in implements most of DITA 1.1 and additional 1.2 features, it is denoted to public domain under the Apache License, Version 2.0.

For more information about I18n Index Library, refer to [http://www.antennahouse.com/product/i18nindexlib\\_V2.1/index.html](http://www.antennahouse.com/product/i18nindexlib_V2.1/index.html) for details.

PDF5 is configured to work without I18n Index Library. If you want use I18n Index Library, refer to use.i18n.index.lib property in • [Plug-in Specific Parameters on page 4](#).

## • PDF5 plug-in features

This plug-in has following features.

### Newly created XSLT2.0 stylesheets for DITA to XSL-FO transformation

This plug-in contains DITA to XSL-FO stylesheets. These stylesheets are written in XSLT 2.0. As DITA Open Toolkit 1.5 and later adopts Saxon-B9.1 as XSLT processor, you can use powerful XSLT 2.0 features in making stylesheets.

### Supports most of DITA elements and attributes for XSL-FO output

The contained stylesheets implements most of DITA 1.1 and additional 1.2 elements and attributes for XSL-FO output. For instance these stylesheets have following features:

- In addition to <indexlist> element, <figurelist>, <tablelist> <glossarylist> elements have been implemented.
- If you use use.oid=yes option, the topic's id is generated using original id and Antenna House Formatter extension attribute for PDF. This makes it possible to link topic from another PDF.
- Supports DITA indexterm features. Multiple level nested <indexterm> elements, multiple <index-see>, <index-see-also> elements, <index-sort-as> element and range index (indexterm/@start, @end attribute) are implemented. About the range index, if the indexterm/@start has no corresponding indexterm/@end element, this stylesheet automatically close the indexterm range according to the DITA specification.
- Implements character base <syntaxdiagram> element and related elements.
- Supports %display-atts; attributes. The %display-atts; attributes contains scale, frame, expanse attributes. You can use these attributes for <fig>, <pre>, <lines>, <codeblock>, <syntaxdiagram>, <properties>, <msgblock>, <simplettable>, <choicetable>, <screen> and <imagemap> elements.

---

## Independent style definition

All of the style are defined in the external file called style definition file in the "config" folder. This style definition file is created for each language-code. Default style definition file, English and CJK style files are bundled in this plug-in. You can change the output document style only editing this style definition file without changing the stylesheets.

## Easy stylesheet customization

If you want to customize the stylesheet algorithms, add the customization stylesheets in the "customization" folder and include it in dita2fo\_custom.xsl. As customization folder is integrated into one folder, you can easily add the customized stylesheet.

- **Limitations**

If this plug-in is used without I18n Index Library, the language support for making index is very limited. The supported language are `xml:lang="en"`, `"ja"` only. If you want to use other language indexing features, please consider to purchase Antenna House I18n Index Library.

In this plug-in the ant build file is made to use Antenna House Formatter to convert XSL-FO to PDF. However if you want to use another Formatter, it is possible to rewrite `com.antennahouse.pdf5\build.xml` to support another one. As this stylesheet uses XSL 1.1 features the target formatter should support XSL 1.1. Please add changes to build file on your responsibility.

---

# Contents

Chapter 1 Plug-in Installation . . . . .	1
Chapter 2 Plug-in And Stylesheet Parameters . . . . .	4
Chapter 3 Command-line Format And Parameter . . . . .	8
Chapter 4 Plug-in process flow. . . . .	9
Chapter 5 Stylesheet Structure. . . . .	11
Chapter 6 Style Definition File . . . . .	14
Chapter 7 Customizing Stylesheet. . . . .	18
Chapter 8 Stylesheet Messages . . . . .	21
Chapter 9 Limitation . . . . .	26

---

# Figures

Figure -1	Integrating PDF5 plug-in into DITA Open Toolkit . . . . .	3
Figure -2	Example of running a batch file in DITA Open Toolkit . . . . .	3
Figure 3-1	Ant command-line formats . . . . .	8
Figure 4-1	Plug-in process flow . . . . .	9
Figure -1	Adding customized stylesheet to dita2fo_custom.xsl . . . . .	18
Figure -2	dita2fo_shell.xsl . . . . .	18
Figure -3	Defining the common graphic file as variable in style definition file . . . . .	19
Figure -4	Referencing logo image path from stylesheet . . . . .	19
Figure -5	Adding PDF5 import instruction in your stylesheet. . . . .	19

---

# Tables

Table 2-1	General plug-in parameters . . . . .	4
Table 2-2	Plug-in specific parameters . . . . .	4
Table 5-1	Stylesheet file and contents . . . . .	11
Table 8-1	Stylesheet messages . . . . .	21

# Chapter 1 Plug-in Installation

This section describes the plug-in installation for the DITA Open Toolkit.

## ■ The plug-in zip file

The plug-in zip file contains this PDF and following file and folders.

```
+---com.antennahouse.pdf5
|   |
|   +--- common-graphic
|   |
|   +--- config
|   |   |
|   |   +--- default_style.xml, en_style.xml, ja_style.xml,
|   |   |       ko_style.xml, zh-CN_style.xml, zh-TW_style.xml
|   |
|   +--- customization
|   |   |
|   |   +--- dita2fo_custom.xsl
|   |
|   +--- xsl
|   |   |
|   |   +--- dita2fo_xxxx.xsl (stylesheet files)
|   |
|   +--- build.xml, integrator.xml, plugin.xml, revision.txt
|
+---index-data
|   |
|   +--- sample_en, sample_ja
|   |   |
|   |   +--- DITA instances and image files
|   |
+---notices
|   |
|   +--- OASIS.txt, PSMT.txt, CommonPublicLicense-v10.html
|
+---test-result
|   |
|   +--- sample_en.pdf, sample_ja_.pdf
|
+--- ahf_setting.xml, run_en.bat, run_ja.bat, README.md,
    pdf5_manual.pdf
```

To install the plug-in into DITA Open Toolkit, follow the next instructions.

---

## ■ Downloading DITA Open Toolkit

You can download DITA Open Toolkit from the following URL: <https://sourceforge.net/projects/dita-ot/files/> There are many kind of archives. The "full\_easy\_install" version is handy because it contains all needed jar files.

After downloading archive file, unzip it into the appropriate folder. From now this folder is called simply [DITA-OT] for convenience.

This plug-in works with DITA-OT 1.6.3 or later.

## ■ Updating DITA Open Toolkit batch file

There is a batch file called startcmd.bat under the [DITA-OT] folder. Add the following two set command before the **start** command line.

```
REM AH Formatter home and setting file
set AHF_DIR=C:\Program Files\AntennaHouse\AHFormatterV61
set AHF_OPT=%DITA_DIR%ahf_setting.xml
start "DITA-OT" cmd.exe
```

The 'AHF\_DIR' environment variable should be changed according to your AH Formatter version. The ahf\_setting.xml is sample option setting file for Formatter. It exists in the archive root folder. Please copy it to the [DITA-OT] folder.

## ■ Installing PDF5 Plug-in

Follow the next instructions.

1. Copy com.antennahouse.pdf5 folder to [DITA-OT]\plugins folder.
2. Copy index-data folder to [DITA-OT]\samples folder.
3. Copy all of the batch file in the root folder to [DITA-OT] folder.
4. Make [DITA-OT]\out folder.
5. From Explorer click [DITA-OT]\startcmd.bat file. The command window titled "DITA-OT" opens.
6. From command window enter the following command. This command integrates PDF5 plug-in into DITA Open Toolkit.

```
ant -f integrator.xml
```



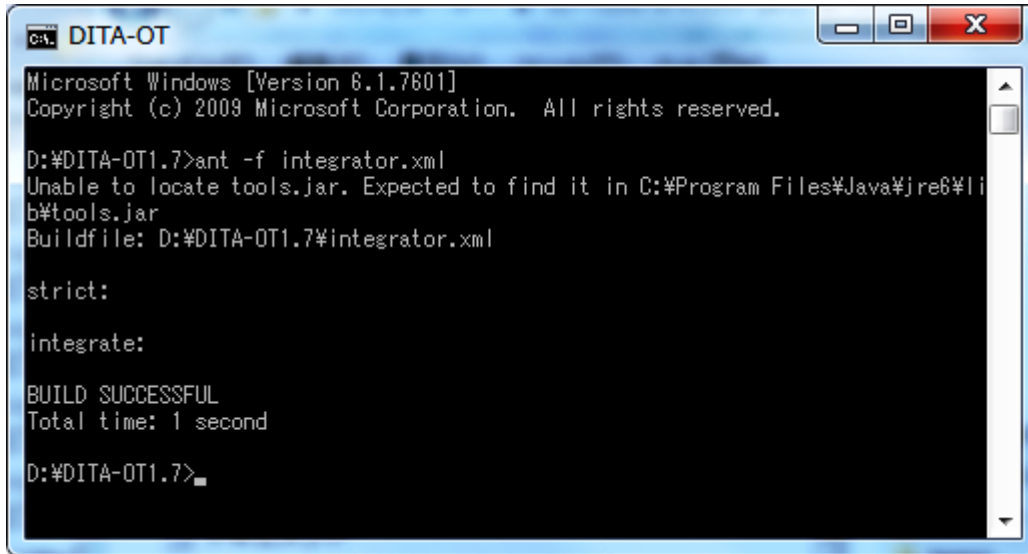


Figure -1 Integrating PDF5 plug-in into DITA Open Toolkit

7. Close the command window.

## ■ Testing PDF5 Plug-in

This plug-in contains test data. Follow the next instructions.

1. From Explorer click [DITA-OT]\startcmd.bat file. The command window titled "DITA-OT" opens.
2. You can test run\_en.bat batch files from this window. This batch file has no parameter.

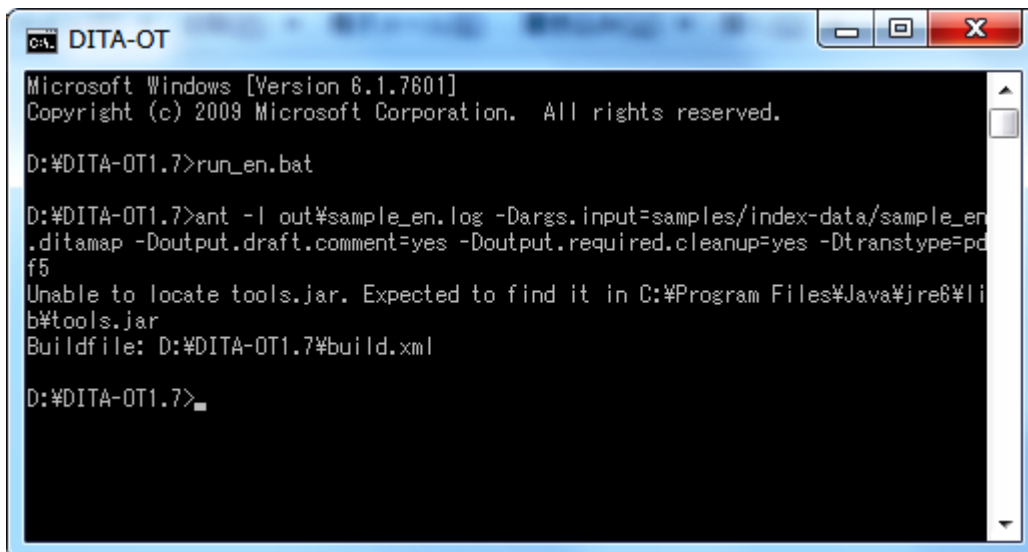


Figure -2 Example of running a batch file in DITA Open Toolkit

The target PDF file and log file will be generated in the [DITA-OT]\out folder.

3. View [DITA-OT]\out\sample\_en.pdf by Adobe Reader.

# Chapter 2 Plug-in And Stylesheet Parameters

This plug-in has plug-in general and plug-in specific command-line parameters.

## • General Plug-in Parameters

Following table describes DITA-OT general plug-in parameters.

Table 2-1 General plug-in parameters

Property name in build file	Name in the stylesheet	Default value	Notes
args.input	-	-	Specify input map file path.
dita.temp.dir	-	-	Specify the folder path where DITA-OT will create temporary files generated during the transformation process.
clean.temp	-	yes	Specifies whether to clean up temporary folder.
dita.input.valfile	-	-	Specifies .ditaval file for filtering.
output.dir	-	-	Specifies the folder path where DITA-OT outputs the results.
transtype	-	-	Specify "pdf5" to invoke this plug-in.

## • Plug-in Specific Parameters

This plug-in and stylesheet has following parameters. The plug-in parameter are defined in the `com.antennahouse.pdf5\build.xml` as property. The stylesheet parameter are defined as `xsl:param` in `com.antennahouse.pdf5\xsl\dita2fo_param.xsl`.

Table 2-2 Plug-in specific parameters

Property name in build file	Name in the stylesheet	Default value	Notes
output.pdf	-	[map file name].pdf	Specifies output PDF file name with extension.
xsl.file	-	com.antenna-house.pdf5\xsl\dita2fo_shell.xsl	Specifies main stylesheet file path.
style.def.file	PRM_STYLE_DEF_FILE	..\config\default_style.xml	Specifies default style definition file.

Property name in build file	Name in the stylesheet	Default value	Notes
alt.style.def.file	PRM_ALT_STYLE_DEF_FILE	..\config\[language-code]_style.xml	Specifies the alternate style definition file path. By default the file name is automatically made from xml:lang attribute value of the map top element.
assume.sortas.pinyin	PRM_ASSUME_SORTAS_PINYIN	no	This parameter is only used with I18n Index Library when making zh-CN manual.
make.see.link	PRM_MAKE_SEE_LINK	yes	Specifies whether to make link for "See" and "See Also" entry in the index page. If this parameter is 'yes', the author must make corresponding indexterm for "See" and "See Also".
include.frontmatter.to.toc	PRM_INCLUDE_FRONTMATTER_TO_TOC	no	Specifies whether to include frontmatter items in the toc page. Usually items in the frontmatter does not appear in the toc.
add.numbering.title.prefix	PRM_ADD_NUMBERING_TITLE_PREFIX	yes	Specifies whether to add numbering to the part, chapter and descending titles. If this parameter is 'yes', the title will be numbered as '1.1.1' style.
add.part.to.title	PRM_ADD_PART_TO_TITLE	yes	Specifies whether to add 'Part' or 'Chapter' to the title when input is described using bookmap. This parameter is available when add.numbering.title.prefix=yes.
add.thumbnail.index	PRM_ADD_THUMBNAIL_INDEX	yes	Specifies whether to make thumbnail index to the right side of the odd page.
xml.lang	PRM_LANG	The xml:lang attribute value of the map top element.	Specifies different xml:lang value for the map.
output.draft.comment	PRM_OUTPUT_DRAFT_COMMENT	no	Specifies whether to output <draftcomment> element content.
output.required.cleanup	PRM_OUTPUT_REQUIRED_CLEANUP	no	Specifies whether to output <required-cleanup> element content.

Property name in build file	Name in the stylesheet	Default value	Notes
use.oid	PRM_USE_OID	no	Decide whether to use original @id value for topic. If the input topics have duplicate @id value, this parameter should be set to "no". If you set this parameter to "yes", you can make external link to this PDF topic from another PDF.
format.dl.as.block	PRM_FORMAT_DL_AS_BLOCK	yes	Specifies whether to format <dl> as block. If this parameter is 'no', <dl> is formatted using table.* <sup>1</sup>
online.pdf	PRM_ONLINE_PDF	yes	If this parameter is 'yes', the stylesheet does not make redundant blank pages for cover, toc, part, chapter and index pages. If this parameter is 'no', the stylesheet forces to the cover, toc, part, chapter and index pages end with even page. As a result the blank page will be inserted before the start page if previous end with odd page.
apply.toc.attr	PRM_APPLY_TOC_ATTR	yes	Specifies whether to honor toc='no' attribute or not.* <sup>2</sup>
use.output.class.deprecated	PRM_USE_OUTPUT-CLASS_DEPRECATED	no	If this parameter is "yes", you can author <ph outputclass="deprecated">. This <ph> is outputted with strikethrough line.
use.output.class.no-hyphenate	PRM_USE_OUTPUT-CLASS_NOHYPHENATE	no	If this parameter is "yes", you can author <table outputclass="nohyphenate"> or <p outputclass="nohyphenate">. This enables to stop hyphenation for the specified element.
output.start.message	PRM_OUTPUT_START_MESSAGE	yes	If this parameter is "yes", plug-in outputs the start message using <xsl:message>.
use.i18n.index.lib	PRM_USE_I18N_INDEX_LIB	no	Decide whether to use I18n Index Library. The

---

Property name in build file	Name in the stylesheet	Default value	Notes
			default value is "no". If you use I18n Index Library, set this parameter to "yes" explicitly.
user.input.dir.url	PRM_MAP_DIR_URL	URL of the input map folder.	Specify input map (book-map) folder URL.

- \*1 Traditionally <dl> is formatted as block. However DITA introduced title element <dlhead>. If you use dl/dlhead element, it is better to format it using table.
- \*2 In the DITA Version 1.1 Architectural Specification p.25, the toc attribute is described to control navigation output. This will be applied for HTML or other output. However current other DITA to XSL-FO stylesheet honors this attribute. This parameter controls that stylesheet should honor toc attribute or not.

---

## Chapter 3 Command-line Format And Parameter

After starting command window by clicking [DITA-OT]\startcmd.bat, you can enter the following format command-line to invoke this plug-in.

```
ant -l [logfile path] -Dtranstype=pdf5 -D[property]=[value] ...
```

Figure 3-1 Ant command-line formats

The [property] is described in the previous section as "Property name in build file". They are almost defined in com.antennahouse.pdf5\build.xml. The file [DITA-OT]\run\_en.bat is a good sample of this command-line.

### Caution

You cannot use Java command-line `java -jar lib/dost.jar ...` to invoke this plug-in. This is because java command-line does not recognize parameters defined for this plug-in.

---

# Chapter 4 Plug-in process flow

This plug-in has four steps to produce PDF from DITA Open Toolkit generated middle file. The simplified diagram is as follows.

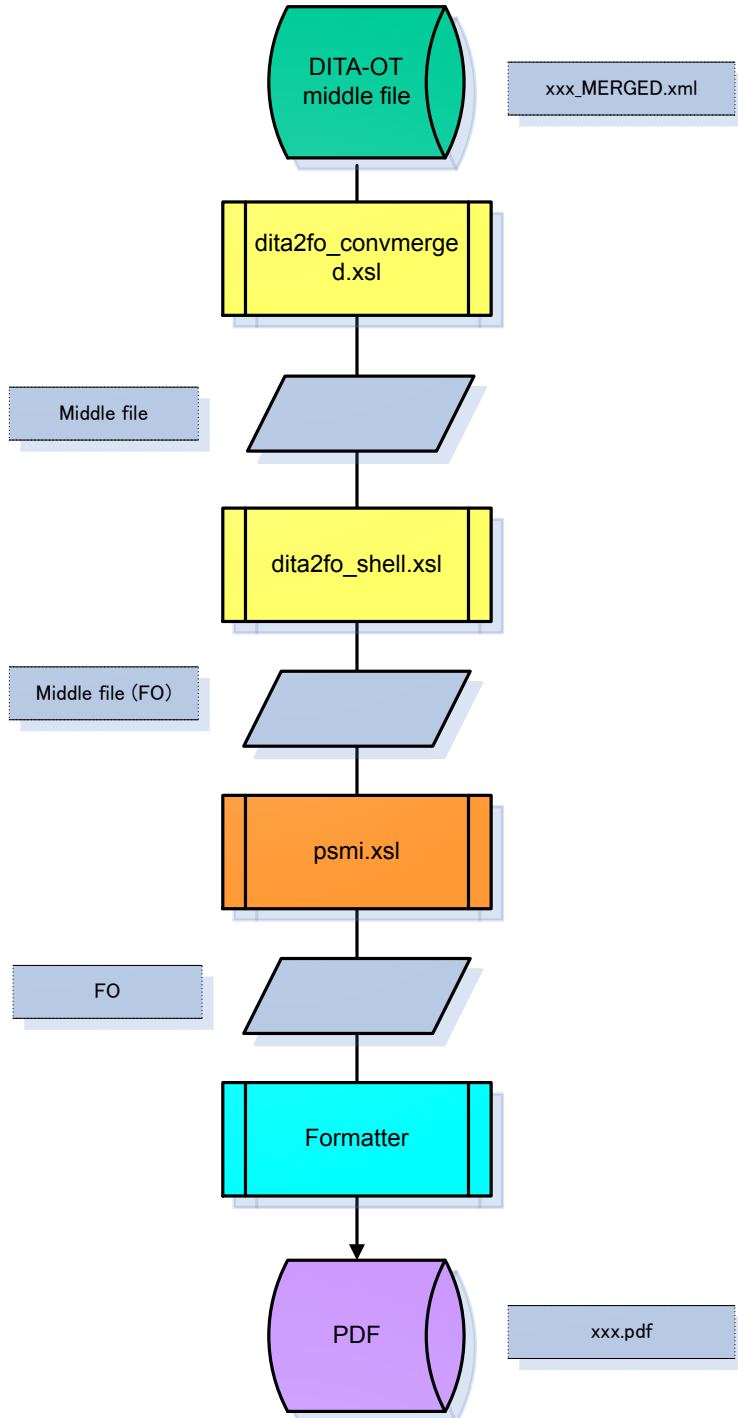


Figure 4-1 Plug-in process flow

---

**xxxx\_MERGED.xml**

This file is merged topic file. All of the topics referenced from the map are merged into one file. This file is located in the [DITA-OT]\temp folder by default. If you want to see this file, add `-Dclean.temp=no` to the command-line parameter.

**dita2fo\_convmerged.xsl**

This stylesheet removes redundant elements from the input file and copy other elements to the output file.

**dita2fo\_shell.xsl**

Main DITA to XSL-FO transformation stylesheet. The main stylesheet file name can be canged by the `-Dxsl.file` command-line parameter.

**psmi.xsl**

This stylesheet adjusts fo:page-sequence for index and toc page.

**Formatter**

Make PDF file from XSL-FO.



# Chapter 5 Stylesheet Structure

The DITA to XSL-FO stylesheets has following names and contents. The stylesheet files are located at `com.antennahouse.pdf5\xsl` folder.

Table 5-1 Stylesheet file and contents

File Name	Contents
dita2fo_attribute.xsl	dita2fo_attribute.xsl Contains %univ-atts;, %display-atts; related function. The id generation is controlled in this stylesheet.
dita2fo_backmatter.xsl	Contains backmatter control templates.
dita2fo_bodyelements.xsl	Contains body related element templates such as <p>, <note>, etc.
dita2fo_bookmark.xsl	This stylesheet makes PDF bookmark.
dita2fo_chapter.xsl	Contains <part>, <chapter> control templates.
dita2fo_common.xsl	Contains common templates and functions.
dita2fo_convmerged.xsl	This stylesheet removes redundant elements from DITA merged middle file. The redundant element contains <topicref> elements that have print='no' attributes, <topicgroup> elements.
dita2fo_cover.xsl	This stylesheet makes cover page.
dita2fo_custom.xsl	This stylesheet is prepared for customization. This stylesheet is located at <code>com.antennahouse.pdf5\customization</code> folder.
dita2fo_documentcheck.xsl	Contains document checking templates.
dita2fo_figurelist.xsl	This stylesheet makes figure list page.
dita2fo_footnote.xsl	This stylesheet handles post-note. This stylesheet does not use standard XSL-FO's footnote. Instead the footnotes are printed as post-note at the end of topic.
dita2fo_frontmatter.xsl	Contains backmatter control templates.
dita2fo_genatrset.xsl	This stylesheet inputs style definition XML file as inner temporary tree.
dita2fo_global.xsl	This stylesheet defines globally used variables.
dita2fo_glossarylist.xsl	This stylesheet handles booklist/glossarylist element and generates glossary list pages.
dita2fo_glossaryrelatedelements.xsl	This stylesheet handles abbreviated-form element.
dita2fo_import.xsl	This stylesheet includes other stylesheets. This stylesheet is imported in dita2fo_shell.xsl.
dita2fo_index.xsl	This stylesheet contains index pre-processing templates that does not use I18n Index Library.
dita2fo_indexcommon.xsl	This stylesheet contains index common templates and index output templates.

File Name	Contents
dita2fo_indexi18n.xsl	This stylesheet contains index pre-processing templates that use I18n Index Library.
dita2fo_indexshell.xsl	This stylesheet controls inclusion of dita2fo_index.xsl, dita2fo_indexi18n.xsl by referencing system property. This stylesheet also includes dita2fo_indexcommon.xsl. The system property is set in the build.xml file.
dita2fo_indexsort.xsl	This stylesheet sorts the expanded indexterm in temporary tree. This stylesheet was originally developed by Dimitre Novatchev. Refer to <a href="http://www.biglist.com/lists/xsl-list/archives/200303/msg00007.html">http://www.biglist.com/lists/xsl-list/archives/200303/msg00007.html</a> for details.
dita2fo_indexterm.xsl	This stylesheet checks <indexterm> element and generates corresponding index-key property or range indexing formatting objects.
dita2fo_layoutmasterset.xsl	This stylesheet generates fo:simple-page-master and fo:page-sequence-master.
dita2fo_main.xsl	This stylesheet contains main control templates.
dita2fo_message.xsl	This stylesheet defines variables for message output.
dita2fo_miscelementelements.xsl	Contains miscellaneous element templates such as <draft-comment>, <fn>, etc.
dita2fo_numberingmap.xsl	This stylesheet makes temporary tree for making <table>, <figure> and <fn> number.
dita2fo_param.xsl	This file defines stylesheet parameter <xsl:param>.
dita2fo_programmingelements.xsl	Contains programming element templates such as <apiname>, <codeblock>, etc.
dita2fo_referenceelements.xsl	Contains reference element templates such as <properties>.
dita2fo_legacyconversionelements.xsl	This styleshet handles required-cleanup element.
dita2fo_relatedlinks.xsl	This stylesheet contains templates for <related-links>. This template does not outputs links for navigation.
dita2fo_shell.xsl	This is entry point stylesheet. Plug-in build file specifies this file as main stylesheet.
dita2fo_softwareelements.xsl	Contains software related element templates such as <msgph>, <msgblock>, etc.
dita2fo_specializationelements.xsl	Contains specialization related element templates such as <itemgroup>, <required-cleanup>, etc.
dita2fo_staticcontent.xsl	This stylesheet generates contents of fo:static-contents.
dita2fo_tableelements.xsl	This stylesheet handles DITA table elements.
dita2fo_tablelist.xsl	This stylesheet makes table list page.
dita2fo_taskelements.xsl	Contains task related element templates such as <cmd>, <info>, etc.

File Name	Contents
dita2fo_thumbindexmap.xsl	This stylesheet makes temporary tree for making thumbnail index.
dita2fo_title.xsl	This stylesheet contains title related templates.
dita2fo_toc.xsl	This stylesheet makes toc page.
dita2fo_topicelements.xsl	Contains topic related element templates such as <abstract>, <shortdesc>, etc.
dita2fo_typographicelements.xsl	Contains typographic related element templates such as <b>, <i>, etc.
dita2fo_userinterfaceelements.xsl	Contains user interface related element templates such as <uicontrol>, <wintitle>, etc.
dita2fo_util.xsl	Contains utility templates and functions.
dita2fo_utilityelements.xsl	Contains utility related element templates such as <imagemap>, etc. Utility elements are used for HTML output.
dita2fo_xref.xsl	This stylesheet contains xref templates.
psmi.xsl	This stylesheet contains templates that adjust fo:page-sequences. Refer to <a href="http://www.cranesoftwrights.com/resources/psmi/index.htm">http://www.cranesoftwrights.com/resources/psmi/index.htm</a> for details. This stylesheet was originally developed by Crane Softwrights Ltd.

# Chapter 6 Style Definition File

Style definition file is a XML file that describes the information of the output PDF layout with XSL stylesheet like notation. You can customize output PDF layout by adding modification to the default style definition file or adding override to the language-specific alternate style definition file.

## ■ Style Definition File

Style definition file is an well formed XML file that have following namespace.

```
http://www.antennahouse.com/names/XSLT/Document/Layout
```

This file has no DTD, you can use following elements in any place.

### <variable>

This element defines the variable. The name attribute is variable name and the child text is the variable value. Variable is referenced from other variable definition or attribute definition. Following is an example of variable definition and reference.

```
<variable name="General_Serif_Font">serif</variable>
<variable name="General_Text_Font">$General_Serif_Font</variable>
```

In this example, the variable value of "General\_Serif\_Font" and "General\_Text\_Font" is both "serif". The "\$" mark is a placeholder of the variable reference.

The variable elements can hold a language specific literals that can be obtained from style-sheet. For example following defines toc, index page title for English.

```
<variable name="Toc_Title">Contents</variable>
<variable name="Index_Title">Index</variable>
```

If the text of variable contains the token %plug-in-path%, it is replaced by the actual plug-in path.

```
<variable name="cNoteIcon">%plug-in-path%common-graphic/note.png</variable>
```

The variable "cNoteIcon" contents will be automatically expanded into file URI "file:/D:/DITA-OT1.7/plugins/com.antennahouse.pdf5/common-graphic/note.png". The value can be obtained using ahf:getVarValue function from the plug-in stylesheet. This notation is useful when you want use plug-in local graphic file which is referenced from the plug-in stylesheet.

### <attribute>

This element defines one attribute. This element has the notation that is mostly like the xsl:attribute element. The name attribute defines attribute's name. The child text is the attribute value. The attribute element must be the child of the attribute-set element. Following is an example of the attribute definition.

```
<attribute name="line-height">normal</attribute>
```

### <attribute-set>

This element defines the attribute set. This element has the notation that is mostly like the `xsl:attribute-set` element. The `name` attribute is the attribute-set's name. The `use-attribute-sets` attribute is used to incorporate the other attribute-set into this attribute-set. The attribute-set element can have the `<attribute>` child elements. Following is an example of the attribute-set definition.

```
<variable name="Base_Font_Size">11pt</variable>
<attribute-set name="atsBaseFontSize">
  <attribute name="font-size">${Base_Font_Size}</attribute>
</attribute-set>
<attribute-set name="atsBaseLineHeight">
  <attribute name="line-height">normal</attribute>
</attribute-set>
<attribute-set name="atsRoot" use-attribute-sets="atsBaseFontSize at
sBaseLineHeight">
  <attribute name="xml:lang">en</attribute>
  <attribute name="font-family">${General_Text_Font}</attribute>
</attribute-set>
```

This example refers variable named "General\_Text\_Font" from font-family attribute definition.

Also this attribute-set incorporates "atsBaseFontSize" and "atsBaseLineHeight" attribute-sets defined in another location.

### <instream-object>

This element defines the svg graphic. This element contains svg namespace child elements. In this stylesheet this svg graphic is used for `<note>` element symbol. If you use svg graphic following namespace should be declared in the style definition file.

```
http://www.w3.org/2000/svg
```

### <formatting-object>

This element defines the XSL Formatting Objects. The name of the object is specified by the `name` attribute. If you use this element following namespace should be declared in the style definition file.

```
http://www.w3.org/1999/XSL/Format
```

### <include>

This element indicates to include another style definition file described in `href` attribute. The `href` attribute has a relative path from the current file.

Refer to the `com.antennahouse.pdf5\config\default_style.xml` file for the actual examples.

---

## ■ How Style Definition File Works

### ● Kinds of style definition file

There are two kind of style definition file in the `com.antennahouse.pdf5\config` folder.

#### Default style definition file

The file name is fixed to `default_style.xml`. This file defines basic variable and styles for English document.

#### Alternate style definition file

The file name is fixed to `[language-code]_style.xml`. The `language-code` is obtained from `xml:lang` attribute of the map root element. If the map root element does not contain `xml:lang` property, `xml:lang="en"` is supposed. This style definition file contains the override for the default style definition file.

### ● Style definition file integration

Prior to the stylesheet document processing the two style definition files are integrated into one temporary tree sequentially. First the default style definition file and next the alternate style definition file will be converted into the temporary tree. In the integration processing the variable reference will be resolved. As a result variable, attribute-set, instream-object, formatting-object temporary tree will be built respectively.

### ● Getting style information

DITA to XSL-FO stylesheets requests the style information by using the name attribute of the `<attribute-set>` element. In this case the attribute-set temporary tree will be searched. If there are multiple attribute-set entry that have the same name, the whole attributes will be returned as the result.

In other words if the same name attribute are defined over the same name attribute-sets in the attribute-set temporary tree, this stylesheet adopts the last defined attribute. By using this mechanism, you can write the language specific style overrides.

### ● Getting the variable value

DITA to XSL-FO stylesheets requests the variable value by using the name attribute of the `<variable>` element. In this case the variable temporary tree will be searched. If there are multiple variable entry that have the same name, the last positioned variable will be adopted.

By using this mechanism, you can write the language specific variables for the various kinds of literal in the output.

### ● Alternate style definition files

There are eight alternate style definition files in the `com.antennahouse.pdf5\config` folder.

---

```
en_style.xml, en-US_style.xml, ja_style.xml, ja-JP_style.xml, ko_style.xml, ko-KR_style.xml, zh-CN_style.xml, zh-TW_style.xml
```

The `en_style.xml` is a dummy file, it has no style definition. The `en-US_style.xml` only includes `en_style.xml`. This file only exists to adopt language code variants in the input map.

### Note

These alternate style definition file contents are not complete yet. You may need to add overrides when you make these language documents.

---

## ■ How To Make Alternate Style Definition File

It is not difficult to make alternate style definition file. For instance, if you want to make a Germany overrides:

1. Make `com.antennahouse.pdf5\config\de_style.xml`
2. Copy XML declaration and the header part from `default_style.xml`.
3. Translate the variable value from English to Germany.
4. If you want to change style, copy the corresponding style definition from `default_style.xml` and change the style value.
5. Specify `xml:lang="de"` to the `ditamap` root element. Then `de_style.xml` will be used in this plug-in.

Refer to bundled alternate style definition file in the `com.antennahouse.pdf5\config` folder for actual examples.

# Chapter 7 Customizing Stylesheet

This plug-in has `com.antennahouse.pdf5\customization` folder for stylesheet customization. In this folder there is `dita2fo_custom.xml`. You can add your customized stylesheet into this file.

## Note

As this stylesheet separates style information as external style definition file, you should customize stylesheet when you want to change the algorithms of template or you want to add a new template for the DITA element, etc. If you want to change the style of the documents, it will be better to use the override mechanism of the style definition file.

## ■ How To Add Customized Stylesheet

If you make a customized stylesheet named `dita2fo_special.xml`, simply add the `xsl:include` instruction to the `com.antennahouse.pdf5\customization\dita2fo_custom.xml`

```
<xsl:stylesheet version="2.0"
xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:axf="http://www.antennahouse.com/names/XSL/Extensions"
xmlns:ahf="http://www.antennahouse.com/names/XSLT/Functions/Document"
exclude-result-prefixes="ahf"
>
<!-- Add your customization xsl here. -->
<xsl:include href="dita2fo_special.xml"/>
</xsl:stylesheet>
```

Figure -1 Adding customized stylesheet to `dita2fo_custom.xml`

As `dita2fo_custom.xml` is included in the shell stylesheet `dita2fo_shell.xml`, this customization will work immediately.

```
<xsl:stylesheet version="2.0"
xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:import href="dita2fo_import.xml"/>
<xsl:strip-space elements="menucascade uicontrol abstract"/>
<xsl:include href="../customization/dita2fo_custom.xml"/>
</xsl:stylesheet>
```

Figure -2 `dita2fo_shell.xml`

## Note

For stylesheet customization mechanisms, refer to `dita2fo_shell.xml`, `dita2fo_import.xml`, `dita2fo_custom.xml`. According to the XSLT import mechanisms, the customization stylesheet is most honored.



## ■ How To Add Logo Image To Output PDF

Usually the image files are referenced from DITA instances by <image> element. They are referenced from output XSL-FO file based on the input map path if it has relative path. However if you want to embed company logo image, it must be treated manually.

To use the local image file simply put them in the `com.antennahouse.pdf5\common-graphic` folder and add variable definition in the style definition file. The style definition processing will automatically expand the variable definition into the file URI if it contains the token "%plug-in-path%". If you put `my-logo.jpg` file in the `com.antennahouse.pdf5\common-graphic` folder, it can be referenced from the stylesheet according to the following example.

```
<variable name="companyLogo">%plug-in-path%common-graphic/my-logo.jpg<variable>
```

Figure -3 Defining the common graphic file as variable in style definition file

```
<xsl:template name="putLogo">
  <xsl:variable name="companyLogoPath" as="xs:string" select="ahf:getVariableValue('companyLogo')"/>
  <fo:block>
    <fo:external-graphic src="{ $companyLogoPath }"/>
  </fo:block>
</xsl:template>
```

Figure -4 Referencing logo image path from stylesheet

## ■ How To Make Your Own Main Customization Stylesheet

This plug-in offers the users to specify the main stylesheet file path from the command-line `-Dxsl.file` parameter. If you make your own main stylesheet, you can freely make your customization stylesheets for your purposes using the existing PDF5 functions. For instance it will be possible to make your own style definition file folder for your each deliverables.

In the main stylesheet you should contain the following imports.

```
<xsl:stylesheet version="2.0"
xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:axf="http://www.antennahouse.com/names/XSL/Extensions"
xmlns:ahf="http://www.antennahouse.com/names/XSLT/Functions/Document"
exclude-result-prefixes="ahf xs"
>
<!-- Import the PDF5 entry. -->
<xsl:import href="../xsl/dita2fo_import.xsl"/>
<!-- Write your own stylesheet code that overrides PDF5 stylesheet. -->
</xsl:stylesheet>
```

Figure -5 Adding PDF5 import instruction in your stylesheet

---

In this case this stylesheet file is located in the some child folder of [DITA-OT]\plugins\com.antennahouse.pdf5.

# Chapter 8 Stylesheet Messages

The DITA to XSL-FO stylesheets outputs the following messages. The suffix character of the message number means:

## The suffix is "I"

This is an information message.

## The suffix is "W"

This is a warning message.

## The suffix is "F"

This is a fatal error message. XSLT processor will stop processing after this message has outputted.

Table 8-1 Stylesheet messages

Message	Contents
[General 001W] No template is defined for this element. element=%elem file=%file	This stylesheet does not have template for %elem. Check the stylesheet file and add template for this element.
[getAttributeSet 005F] Attribute-set name not found attribute-set=%attrsetname file=%file	Check the style definition file=%file.
[getAttributeValue 006F] Attribute-set name not found. attributeset=%attrsetname file=%file	Check the style definition file=%file.
[getVarValue 008F] Variable %var is not found in %file.	Check the style definition file=%file.
[getInstreamObject 009F] Instream object name not found. objname=%objname file=%file	Check the style definition file=%file.
[getFormattingObject 010F] Formatting object name not found. obj-name=%objname file=%file	Check the style definition file=%file.
[getVarRecursive 023F] Referenced variable %var-name is not exist in %stylefile	Check the style definition file=%stylefile.
[getCssStyle 025F] Style %style does not defined in %file.	Check the style definition file=%file. This is inner error.
[percentToNumber 028W] Invalid @scale value. Assumed 100. scale=%scale element=%elem file=%file	Correct the authoring.
[processLocalXref 030F] Xref/@href destination topic does not found. Probably href format is illegal. href=%h file=%file.	Correct the xref/@href authoring.
[getXrefTitle 031W] Xref/@href destination section has no title element. Section id=%id file=%file	Correct the xref or section authoring.
[getXrefTitle 032W] Xref/@href destination example has no title element. Example id=%id file=%file	Correct the xref or example authoring.

Message	Contents
[getXrefTitle 033W] Xref/@href destination table has no title element. Table id=%id file=%file	Correct the xref or table authoring.
[getXrefTitle 034W] Xref/@href destination fig has no title element. Fig id=%id file=%file	Correct the xref or fig authoring.
[getXrefTitle 035W] Xref/@href destination %elem has no title content. %elem id=%id file=%file	Correct the xref authoring.
[getFootnotePrefix 037W] The fn element does not have table,simpletable,ul,ol,dl,glossdef as ancestor. This fn will be ignored. Content='%cont' file=%file	Correct the fn authoring.
[bodyelements 041W] Element object is not suitable for PDF output. Ignored. file=%file classid=%class	Correct the object authoring.
[getKeyCol 050W] The keycol attribute is not positive integer. Assumed as not specified. file=%file element=%elem keycol=%keycol	Correct the @keycol authoring.
[dlhead 055W] As PRM_FOR-MAT_DL_AS_BLOCK='yes', the dl/dlhead element is ignored. file=%file	Change the parameter PRM_FOR-MAT_DL_AS_BLOCK='no' or delete dl/dlhead element.
[synnoteref 060W] The href attribute of synnoteref cannot be resolved. file=%file trace=%trace @href=%href	Check the synnoteref authoring.
[processLink 062W] Ignored invalid link in related-links or reltable. file=%file href=%href	Check the reltable or relatedlinks authoring.
[processLink 063W] Ignored invalid link in related-links or reltable. file=%file href=%href	Check the reltable or relatedlinks authoring.
[processLink 070W] topicref/@href target does not found. href='%href' file=%file	Check the topicref authoring.
[processLocalXref 072F] Xref target is not contained in map. href='%href' file=%file	Check the xref target.
[makeBasicLinkDestination 074F] Topic in href target does not found. href='%href' file=%file element=%element	Check the href attribute of the specified element.
[processTopicref 076F] topicref/@href target does not found. href='%href' file=%file	Check the href attribute of the topicref element.
[genGlossaryListMain 078F] Referenced topicref does not found. id of topic='%id' file=%file	Check the corresponding glossentry element.
[genGlossaryListBookMark 079F] Referenced topicref does not found. id of topic='%id' file=%file	Check the corresponding glossentry element.
[Frontmatter 080W] Element:%elem without @href attribute is ignored. file=%file	Check the corresponding element.
[Frontmatter 082W] Element:%elem without @href attribute is not supported. file=%file	Check the corresponding element.
[Backmatter 084W] Element:%elem without @href attribute is ignored. file=%file	Check the corresponding element.

Message	Contents
[Backmatter 086W] Element:%elem without @href attribute is not supported. file=%file	Check the corresponding element.
[Figurelist 087W] Element:%elem is ignored. There is no figure with title. file=%file	Check the corresponding element.
[Tablelist 088W] Element:%elem is ignored. There is no table with title. file=%file	Check the corresponding element.
[Glossarylist 089W] Element:%elem is ignored. There is no glossentry as the child of %elem. file=%file	Check the corresponding element.
[Glossarylist 090W] Element:%elem should be written in backmatter otherwise all of indexterm after the %elem will be automatically ignored. file=%file	Check the corresponding element.
[ditamapClass 100F] Undefined ditamap class. class=%class file=%file	Check the map file.
[documentLang 101W] xml:lang is not specified in map. Adopt 'en' as default language.	Check the ditamap file.
[altstyledef 102W] Alternate style definition file %file does not found. Stylesheet use %default as style definition file.	Check the file specified alt.style.def.file parameter of command-line.
[altstyledef 103W] Alternate style definiton file %file does not found. Stylesheet use %default as alternate style definition file.	Check the xml:lang attribute of map root element or check the well-formedness of the alternate style definition file.
[styledef 104F] Style definiton file %file does not found.	Check the file specified style.def.file parameter of command-line.
[getPropertyNu 400W] Invalid non-numeric property value=%val'. Treated as 1.0.	Some font-size attribute in the style definition file is not numeric.
[documentCheck 500F] topicref/@href target does not found. ditamap=%xtrf href=%ohref	Check the ditamap file.
[documentCheck 501F] Detected same topic/@id value. It must be unique. @id=%id' file1=%xtrf1 file2=%xtrf2	Correct the id value of the topic.
[documentCheck 503F] The part and chapter level element coexists. Either part or chapter only document is supported. ditamap=%xtrf	This stylesheet does not accept this configuration. Correct the ditamap file.
[documentCheck 504F] The part level element contains attribute toc='no'. file=%ohref ditamap=%xtrf	This stylesheet does not accept this pattern. Correct the ditamap file.
[documentCheck 505F] The chapter level element contains attribute toc='no'. file=%ohref ditamap=%xtrf	This stylesheet does not accept this pattern. Correct the ditamap file.
[documentCheck 514F] This plug-in does not support a indexlist element in frontmatter. ditamap=%xtrf	Correct the authoring.
[documentCheck 599F] Terminated by document fatal error.	Check the previous error message.
[genIndex 600F] Failed to sort indexterm. Counts before sorting= %before, after sorting=%after	This will be the difficulty of the I18n Index Library.
[genIndex 601I] Index debug start.	This is debug message.

Message	Contents
[genIndex 602I] Index debug end.	This is debug message.
[indexterm 610W] Authoring error! Indexterm has sibling index-see element. index-key='%key' file=%file Stylesheet will ignore this index-see element.	Correct the indexterm authoring.
[indexterm 611W] Authoring error! Indexterm has sibling indexsee-also element. index-key='%key' file=%file Stylesheet will ignore this index-see-also element.	Correct the indexterm authoring.
[indexterm 612W] Authoring error! Indexterm has both index-see and index-see-also child element. index-key='%key' file=%file Stylesheet will ignore this index-see element.	Correct the indexterm authoring.
[indexterm 620W] Authoring error! Text of indexterm is empty. index-key='%key' file=%file Stylesheet will ignore this indexterm element.	Correct the indexterm authoring.
[indexterm 621W] Authoring error! Text of indexterm is too long. It must be less than %max characters for sorting. index-key='%key' file=%file Stylesheet will trim this indexterm text for index sorting.	Correct the indexterm authoring. This message will not output when you use I18n Index Library.
[indexterm 630W] Authoring error! Start attribute is specified in nesting indexterm more than once. previous='%prev' current='%curr' index-key='%key' file=%file Stylesheet will ignore this start attribute.	Correct the indexterm authoring.
[indexterm 640W] End attribute authoring error! Corresponding indexterm that has same start attribute value does not found. end='%end' index-key='%key' file=%file Stylesheet will ignore this end attribute.	Correct the indexterm authoring.
[indexterm 641W] End attribute authoring error! Corresponding indexterm that has same start attribute value exist plurally. end='%end' index-key='%key' file=%file Stylesheet will ignore this end attribute.	Correct the indexterm authoring.
[indexterm 642W] Authoring warning! Indexterm element that has end attribute should not have ancestor indexterm element. end='%end' index-key='%key' file=%file Stylesheet will ignore ancestor indexterm.	Correct the indexterm authoring.
[indexterm 643W] Authoring warning! Indexterm element that has end attribute should not have child indexterm element. end='%end' index-key='%key' file=%file Stylesheet will ignore child indexterm.	Correct the indexterm authoring.
[indexterm 650W] Authoring error! Attribute start and end cannot be specified together in one indexterm. start='%start' end='%end' index-key='%key' file=%file Stylesheet will ignore this indexterm.	Correct the indexterm authoring.
[indexterm 651W] Authoring error! End attribute cannot be specified to the descendant of indexterm that has start attribute. start='%start' end='%end' index-key='%key' file=%file Stylesheet will ignore this indexterm.	Correct the indexterm authoring.

Message	Contents
[indexterm 652W] Authoring error! Start attribute cannot be specified to the descendant of indexterm that has end attribute. start='%start' end='%end' index-key='%key' file=%file Stylesheet will ignore this indexterm.	Correct the indexterm authoring.
[makeChapterMap 700W] Illegal class is found in topicref. class='%class' file=%file.	Check the ditamap file.
[attribute-set 900F] Illegal attribute found in attribute-set element in style definition. attribute-set-name='%attribute-set-name' attribute='%attribute'.	Check the attribute of attribute-set element in the style definition file.
[attribute-set 902F] Attribute-sets references itself by @use-attribute-sets attribute. attribute-set-name='%attribute-set-name'.	Check the use-attribute-sets attribute of attribute-set element in the style definition file.
[convmmerged 1001I] Removing topicref. href=%href ohref=%ohref	An topicref element is removed because it has @print="no" attribute.
[convmmerged 1002I] Removing topic. id=%id xtrf=%xtrf	An topic element is removed because it is referenced from topicref that has @print="no" attribute.
[convmmerged 1003I] Removing link. href=%href	An link element is removed because it refers the removed topic.
[convmmerged 1004W] Warning! Xref refers to removed topic. href=%href	An xref element refers removed topic.

---

# Chapter 9 Limitation

This plug-in has following important limitations.

## Reference to the nested topic.

The following `topicref/@href` is not supported in a map (without `reltable`).

```
<topicref href="abc.xml#topicId">
```

where "topicId" is not the top level topic/@id of abc.xml.

## <fn> element does not generate <fo:footnote> defined in XSL-FO specification.

This plug-in assumes that `<fn>` element exist in `<table>`, `<simplettable>`, `<ul>`, `<ol>`, `<dl>`, `<glossdef>` elements and outputs the `<fn>` contents below of these elements. The `<fn>` element in other elements will be ignored.

Specifying parameter `$PRM_DISPLAY_FN_AT_END_OF_TOPIC="yes"` will outputs the `<fn>` contents at the end of topic. (This parameter is deprecated and not included in build.xml.)