Antenna House I18n Index Library V2.3

Mar, 2016. Antenna House, Inc.

Revision

Date	Contents
Jun 04, 2009	Newly published.
Nov 20, 2009	Add V2.1 description.
Jan, 2014	Revised for V2.2 release.
Mar, 2016	Revised for V2.3 release.

Contents

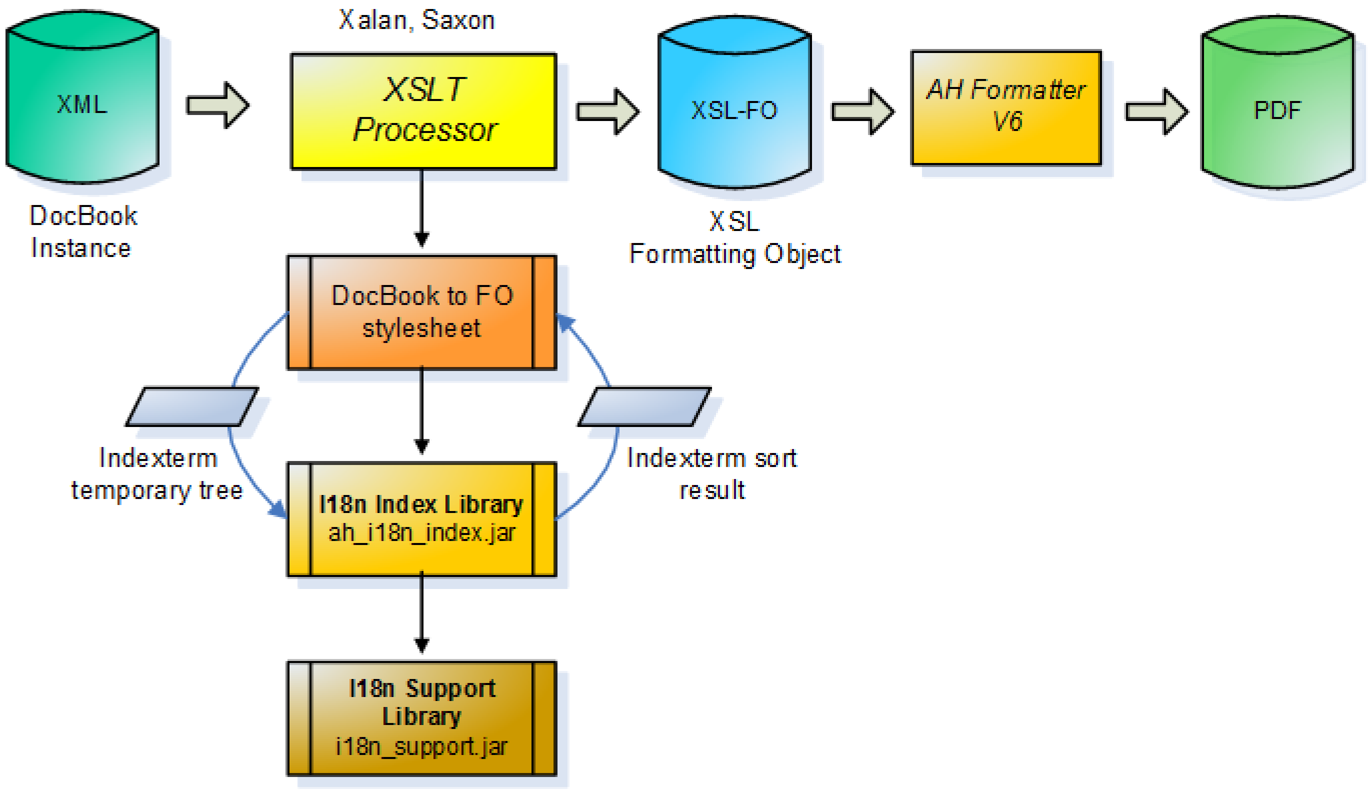
1 Overview

Antenna House I18n Index Library is a Java library that makes index pages in various languages used by the DocBook, DITA to XSL-FO stylesheet. Also it offers language specific sorting capability that handles any sort of element.

1.1 Processing diagram

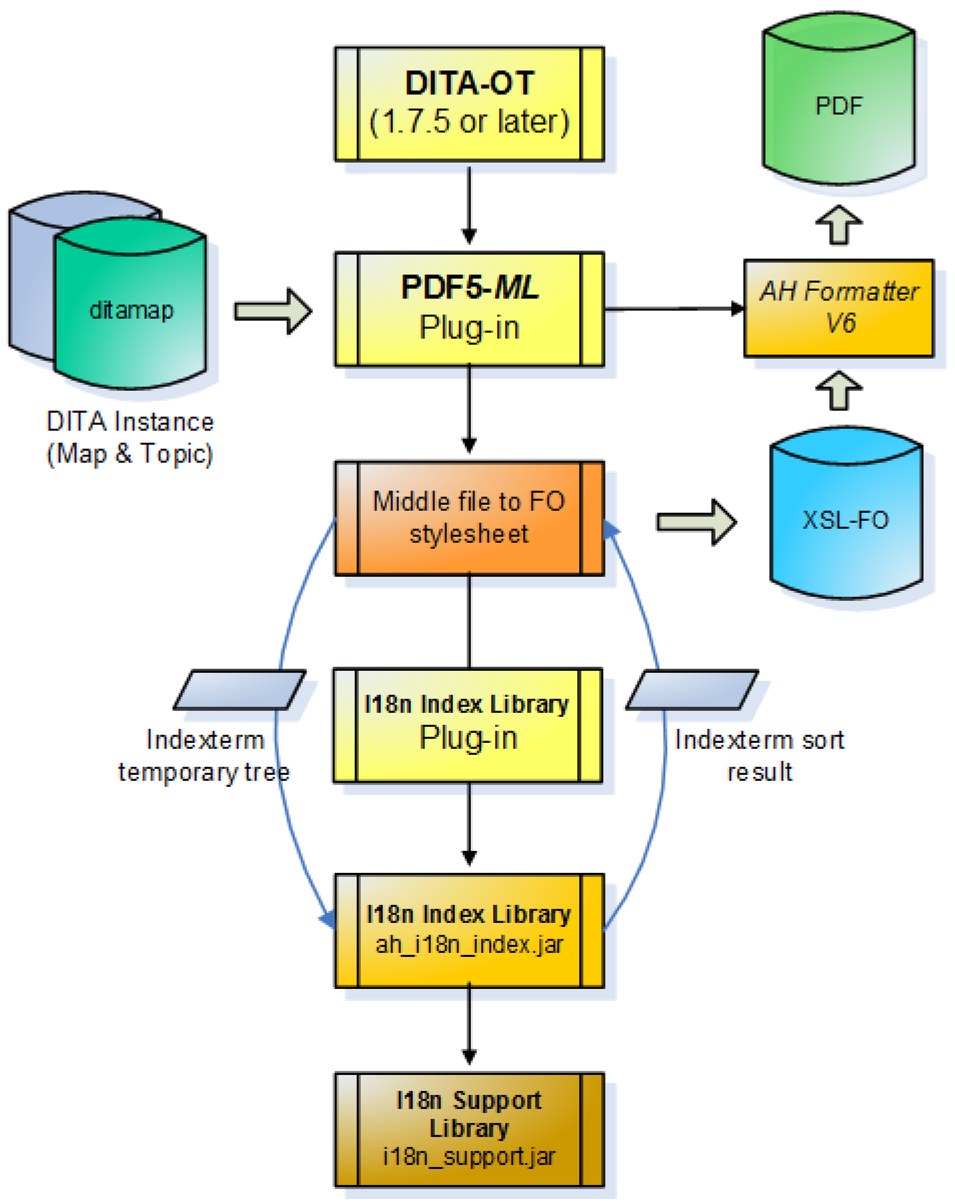
The processing diagrams for using the I18n Index Library are as follows:

[Fig.1] Processing diagram for DocBook



- I18n Support Library is the LGPL open-source license library developed by Innodata Isogen, Inc. Antenna House made some modifications to the original library and opened this library under the LGPL license according to the LGPL rule. This library is included in this release.
- I18n Index Library is an interface library between the stylesheet and the I18n Support Library developed by Antenna House. It provides the sorting capability for the supported language indexes.
- The DocBook to FO stylesheet is a sample stylesheet developed by Antenna House. This stylesheet passes index data to the library and makes the formatting objects for indexing from the sorting results. This release contains XSLT 1.0 and XSLT 2.0 stylesheets for convenience. You can modify this stylesheets according to your PDF output requirements.
- AH Formatter outputs PDF from XSL-FO and it implements XSL 1.1 indexing features. Antenna House Formatter V6 is recommended.

[Fig.2] Processing diagram for DITA Open Toolkit



- The DITA Open Toolkit (DITA-OT) is an open-source publishing system for XML instances written in DITA.
- PDF5-ML is a sample DITA-OT plug-in developed by Antenna House. It implements most of DITA 1.2 index features and outputs PDF using Antenna House Formatter.
- I18n Index Library plug-in is an independent DITA-OT plug-in that enables multiple DITA-OT plug-ins to use the I18n Index Library feature.

1.2 Highlights

1.2.1 Language support

- Supports 50 language indexes.

Arabic, Bulgarian, Catalan, Czech, Danish, German, Greek, English, Spanish, Estonian, Persian (Farsi), Finnish, French, Hebrew, Hindi, Croatian, Hungarian, Indonesian, Italian, Icelandic, Japanese, Kazakh, Khmer, Kannada, Korean, Lao, Lithuanian, Latvian, Malay, Burmese (Myanmar), Dutch, Norwegian, Polish, Portuguese, Romanian, Russian, Sinhala, Slovak, Slovenian, Swedish, Swahili, Tamil, Teglu, Thai,

Tagalog, Turkish, Ukrainian, Vietnamese, Simplified Chinese, Traditional Chinese

NOTE: *The corresponding language codes for supported languages are ar, bg, ca, cs, da, de, el, en, es, et, fa, fi, fr, he, hi, hr, hu, id, it, is, ja, kk, km, kn, ko, lo, lt, lv, ms, my, nl, no, pl, pt, ro, ru, si, sk, sl, sv, sw, ta, te, th, tl, tr, uk, vi, zh-CN, zh-TW*

- Supports derivatives for language codes.

- For instance you can define one index configuration for pt, pt-BR, pt-PT.
- These language codes can be written in index configuration file botb_index_rules.xml as follows.

```
<index_config>
    <national_language>pt</national_language>
    <national_language>pt-PT</national_language>
    <national_language>pt-BR</national_language>
    <description>
        <p>Portuguese index configuration</p>
    </description>
    ...
<index_config>
```

1.2.2 DocBook to FO stylesheet

- Implements many DocBook indexterm features using XSL1.1 index functions.

- Three-level nested index structure (primary, secondary and tertiary elements).
- Range index (startofrange, endofrange attribute).
- Significant index (significance attribute).
- Supports "See", "See Also" (see, seealso elements).

- Supports major Java based XSLT processors.

- XSLT 1.0 processor: Saxon 6.5.5, Xalan-J 2.7.1/2.7.2
- XSLT 2.0 processor: Saxon-B 9.1, Saxon-PE/EE 9.2 or later

- Includes sample XSLT 1.0/2.0 based stylesheets.

- XSLT 1.0 stylesheets for Saxon 6.5.5, Xalan-J 2.7.1/2.7.2
- XSLT 2.0 stylesheets for Saxon-B 9.1, Saxon-PE/EE 9.2 or later

- You can use the sortas attribute to correct Simplified Chinese index orders.

For example the Chinese word "粘贴" belongs to "N" index group because the most common reading of "粘" is "nian2". However the correct reading is "zhan1" for this word.

```
<indexterm><primary>粘贴</primary></indexterm>
```

You can correct this problem by specifying the correct reading (pinyin) to the sortas attribute value. The fix will place "粘贴" into the "Z" index group.

```
<indexterm><primary sortas="zhan1 tie1">粘贴</primary></indexterm>
```

1.2.3 PDF5-ML plug-in

- Supports DITA indexterm features.

- Multiple level nested <indexterm> elements.
- Multiple <index-see>,<index-see-also> elements.
- <index-sort-as> element.
- Range index (indexterm/@start, @end attribute).

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

- Supports many DITA 1.3 elements and attributes. For instance, in addition to <indexlist> element, <figurelist>, <tablelist> elements have been implemented.
- All of the style are defined in the external file called style definition file in the "config" directory. This style definition file is created for each language-code. Default style definition file, English and CJK files are bundled in this plugin. You can change the manual style only editing this style definition file without editing stylesheet file.
- If you want to customize the stylesheet algorithms, add the customization stylesheets in the "customize" folder and include it in dita2fo_custom.xsl. You can use full power of XSLT 2.0 features for DITA to FO transformation.
- If indexterm/@start has no corresponding indexterm/@end element, this stylesheet automatically close the indexterm range according to the DITA specification. This process is done at topicref/topicmeta, topic/metadata and body-level range indexterm elements.

NOTE: *PDF5-ML plug-in is independent from this I18n Index Library release. It can be downloaded from GitHub https://github.com/AntennaHouse/pdf5-ml*

NOTE: *The old PDF5 plug-in is also available in GitHub https://github.com/AntennaHouse/pdf5. However PDF5 is no more maintained because the successor PDF5-ML has been developed.*

1.2.4 I18n Index Library plug-in

- Independent Java library plug-in

- I18n Index Library plug-in can be shared from another plug-ins in DITA Open Toolkit . For instance it can be used from PDF, HTML, EPUB plug-ins if they need index sorting function.
- Installation is very easy. You can copy I18n Index Library plug-in folder into `[DITA-OT]\plugins` folder and integrate it using ant command-line.

1.2.5 General element sorting function

- Sorts any kind of elements

- I18n Index Library has one more class named jp.co.antenna.ah_i18n_generalsort. You can use static method of this class for sorting any kind of elements.
- The sort key should be supplied by @sort-key and @sort-as attribute of the target element.

- Differences between <xsl:sort> and I18n Index Library

- By calling static method of jp.co.antenna.ah_i18n_generalsort class you can group sorting results by @group-key attribute that library returns.
- Language specific grouping cannot be done with standard <xsl:sort>.

2 System Requirements

Item	Contents
Java Runtime Environment	Oracle JRE 7 or later
XSLT Processor	Saxon 6.5.5, Xalan-J 2.7.1/2.7.2, Saxon-B 9.1, Saxon-PE/EE 9.2 or later
XML Parser	Java bundled parser
DITA Open Toolkit	1.7.5 or later
XSL Formatter	Antenna House Formatter V6

NOTE:

- *Saxon-HE 9.2 or later are not supported because they do not allow external Java library call from XSLT stylesheet.*
- *DITA Open Toolkit 1.7.5 or later bundles Saxon-B 9.1 as XSLT Processor.*

3 Limitations

- **This library supports Unicode characters only in the BMP (Basic Multilingual Plane) for the sorting index.** The Hanzi or other characters that are outside the BMP are not supported.
- This release uses sample batch files which only operate in a Windows environment.
- The DocBook zone and the startref attributes of the indexterm element are not supported.
- I18n Index Library V2.3 is tested via ICU4J 5.6 release (icu4j-56_1.jar). Refer to ICU site for details.
- Hanzi collation is made from Unicode 6.0 Unihan databases. Refer to Unihan Database for details.
- The sorting architecture between PDF5-ML plug-in and PDF2 plug-in bundled with DITA-OT is significantly different. The operation integrating I18n Index Library into PDF2 plug-in is not tested nor officially supported.

NOTE: *DITA-OT bundles its own ICU library. If you use DITA-OT, this ICU library is given priority over the ICU library bundled with I18n Index Library because preloaded class takes precedence. For instance DITA-OT 2.2.2 bundles icu4j-54.1.jar. The sorting result may slightly differ due to the version of the ICU library.*

4 Getting Started With DocBook Sample

This section describes how to make a sample PDF using the included batch files.

4.1 Choosing an XSLT Processor

Select the XSLT processor from Saxon 6.5.5, Xalan-J 2.7, Saxon-B 9.1, Saxon PE/EE 9.2 or later and then download it from the following URL.

XSLT Processor	URL	Comments
Saxon 6.5.5	http://saxon.sourceforge.net/	
Xalan-J 2.7.1/2.7.2	http://xml.apache.org/xalan-j/	
Saxon-B 9.1	http://saxon.sourceforge.net/	
Saxon-PE/EE 9.2 or later	http://www.saxonica.com/	Saxon-PE and EE are the commercial products.

After downloading the archive, unzip it into an appropriate folder.

4.2 Updating Batch File

Modify the `docbook\startcmd.bat` batch file according to the selected XSLT processor. Rewrite the following values to correspond to the location where the .jar file was saved. Not all XSLT processors are needed.

```
SET SAXON6_HOME=%PROJECT_HOME%xslt\saxon6-5-5
SET XALAN2_HOME=%PROJECT_HOME%xslt\xalan-j_2_7_1
SET SAXON9_HOME=%PROJECT_HOME%xslt\SaxonPE9-7-0-2J
```

For instance, if you want to test using Saxon9 only, rewrite SAXON9_HOME environmental variable and comment out the other lines.

```
REM *** Example ***
REM SET SAXON6_HOME=%PROJECT_HOME%xslt\saxon6-5-5
REM SET XALAN2_HOME=%PROJECT_HOME%xslt\xalan-j_2_7_1
SET SAXON9_HOME=C:\MY_DOWNLOAD\xslt\SaxonbPE9-7-0-2j
```

Next, rewrite the following line according to your Formatter environment.

```
REM AHF Formatter home
SET AHF_HOME=C:\Program Files\Antenna House\AHFormatterV62
```

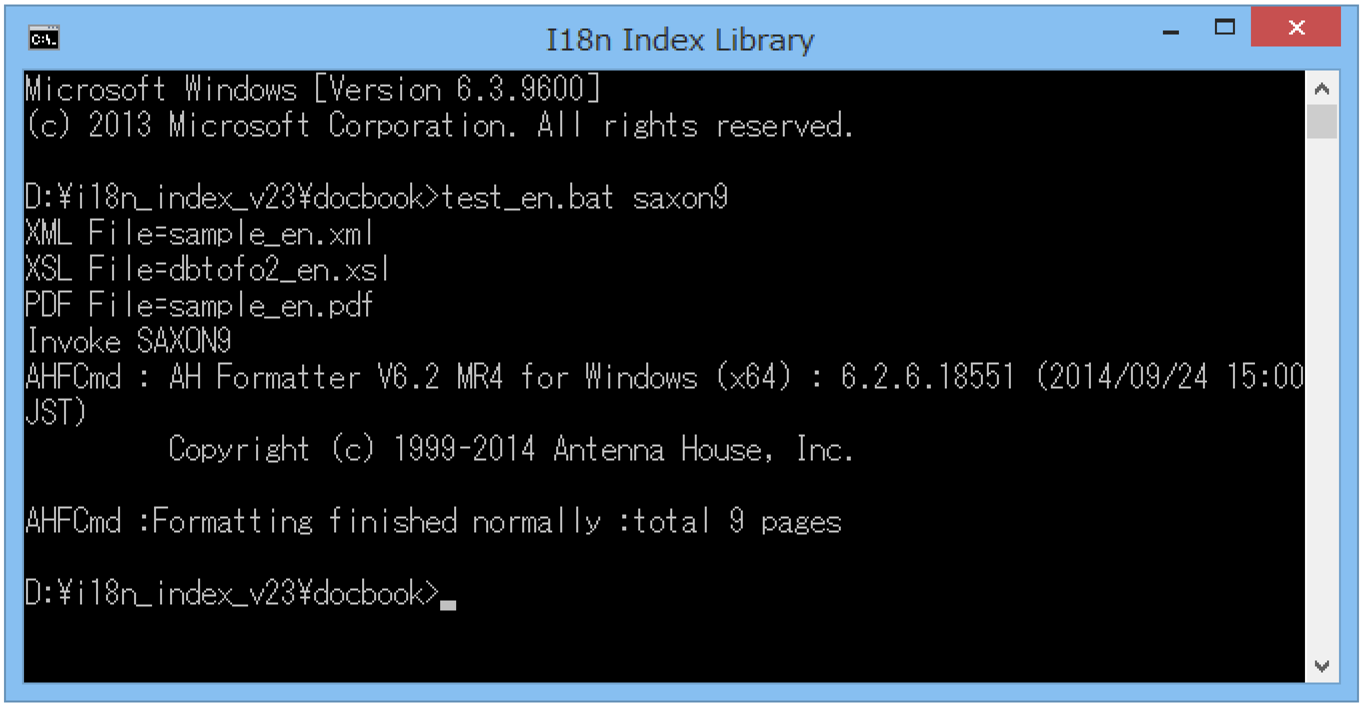
4.3 Testing PDF Output

1. From Explorer, click `docbook\startcmd.bat`. The command window titled "I18n Index Library" opens.
2. You can test following batch files from this window.
    - test_en.bat
    - test_zhcn_normal.bat
    - test_zhcn_sortaspinyin.bat
    - test_ja.bat
    - test_ko.bat

   The command format is as follows. Select "saxon6" ,"xalan2" or "saxon9" as parameter according to the XSLT processor that you downloaded.

   ```
   test_en.bat [saxon6 | xalan2 | saxon9]
   ```

[Fig.3] Example of running a batch file

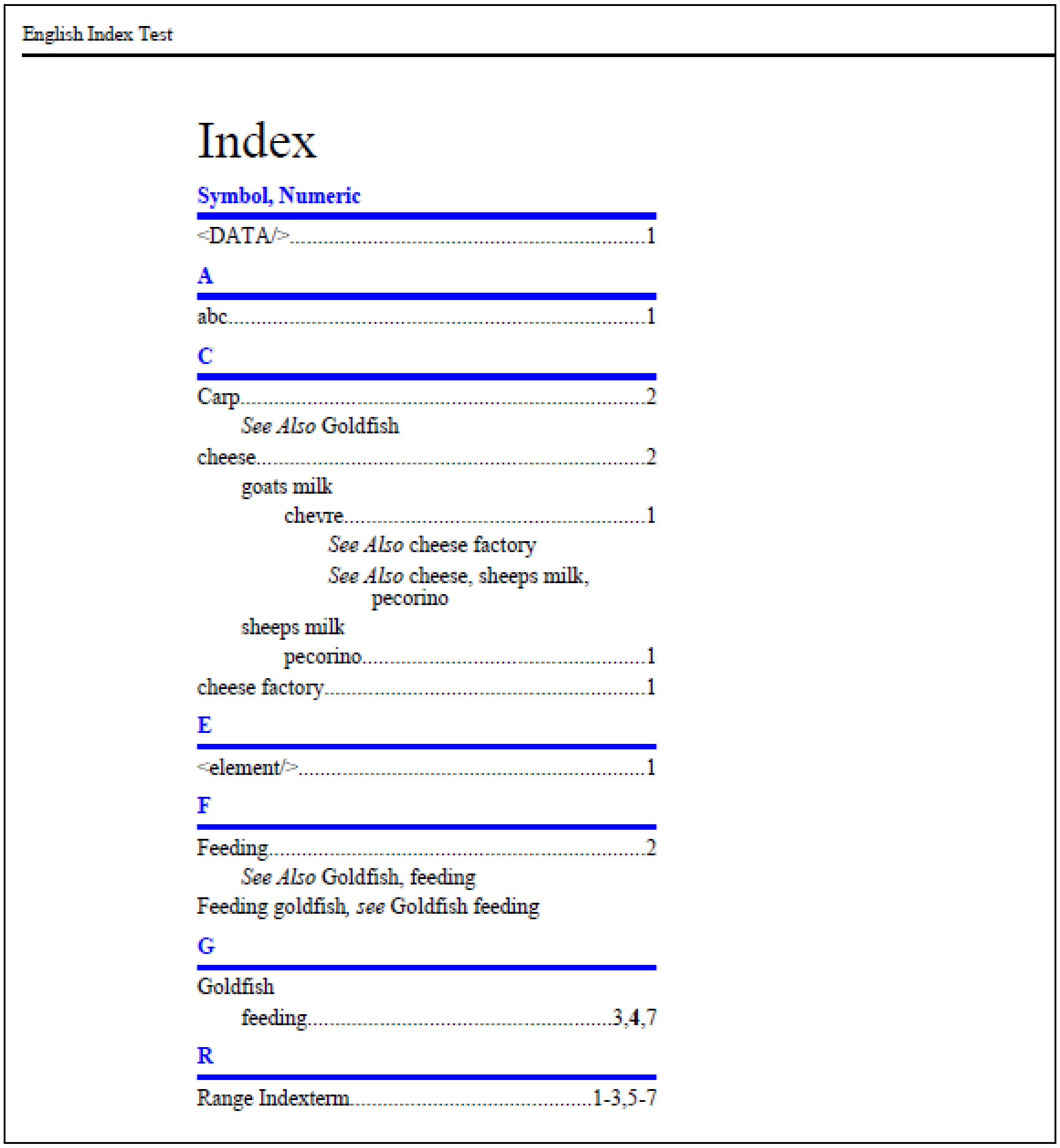


I18n Index Library

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

D:¥i18n_index_v23¥docbook>test_en.bat saxon9
XML File=sample_en.xml
XSL File=dbtofo2_en.xsl
PDF File=sample_en.pdf
Invoke SAXON9
AHFCmd : AH Formatter V6.2 MR4 for Windows (x64) : 6.2.6.18551 (2014/09/24 15:00
JST)
         Copyright (c) 1999-2014 Antenna House, Inc.

AHFCmd :Formatting finished normally :total 9 pages

D:¥i18n_index_v23¥docbook>_
```

The result of sample_en.pdf is saved to the "out" folder.

[Fig.4] English output PDF sample


English Index Test

Index

Symbol, Numeric

<DATA/>....1

A

abc....1

C

Carp....2
    See Also Goldfish
cheese....2
    goats milk
        chevre....1
            See Also cheese factory
            See Also cheese, sheeps milk, pecorino
    sheeps milk
        pecorino....1
cheese factory....1

E

<element/>....1

F

Feeding....2
    See Also Goldfish, feeding
Feeding goldfish, *see* Goldfish feeding

G

Goldfish
    feeding....3,4,7

R

Range Indexterm....1-3,5-7

4.4 Test Data Features

By running the batch files, you can confirm the following features:

Batch File	PDF File	Feature	Used Font
test_en.bat	sample_en.pdf	DocBook and XSL 1.1 features; • Three level nested indexterm. (primary, secondary, tertiary elements) • See and See Also feature. • Change index position using sortas attribute. (Sample: <element/>)	Times New Roman

		Significant="preferred" attribute. (Reference page is displayed as bold: "3, **4**, 7") • Range indexterm.(Range is displayed using hyphen: "1-3, 5-7")	
test_zhcn_normal.bat	sample_zhcn_normal.pdf	Simplified Chinese (zh-CN) index feature; Index is sorted by following keys and grouped by the first character of the pinyin. `Pinyin/Strokes/Radical/GB2312-90 Code Order`	SimHei
test_zhcn_sortaspinyin.bat	sample_zhcn_sortaspinyin.pdf	Simplified Chinese index correction using sortas attribute feature; As Chinese Hanzi has multiple readings sometimes the index entry is not placed into the correct index group. To solve this problem we set the correct reading to the sortas attribute. By doing this, the following words will be placed into the correct index group position. • **重**选择 (zhong4→chong2) Reselection • 汉**字支持** (yi4→han4) Hanzi support • **泊松到达** (bo2→po1) Poisson arrival • 粘贴 (nian2→zhan1) Paste For details, see sample-zhcn2.xml file.	SimHei
test_zhtw.bat	sample_zhtw.pdf	Traditional Chinese index feature; The index is sorted by the following keys and grouped by strokes. `Strokes/Radical/Big5 Code Order`	MingLiu
test_ja.bat	sample_ja.pdf	Japanese index feature; The Japanese index uses sortas for its readings. Index are sorted by sortas order and grouped by consonants. For more details see original XML file sample_ja.xml.	MS Mincho
test_ko.bat	sample_ko.pdf	Korean index feature; The Korean index is sorted by its syllable block order and the index is grouped by	Batang

		the consonant of the first character.	

4.5 Customizing Stylesheet

This library has XSLT1.0 and XSLT2.0 sample stylesheets in the stylesheet folder. These stylesheets create XSL-FO from DocBook instance. Each stylesheet file has the following role. (XSLT1.0 stylesheet has the prefix "dbtofo". XSLT2.0 stylesheet has the prefix "dbtofo2".)

File	Contents
dbtofo.xsl	Shell stylesheet that includes other needed stylesheet files.
dbtofo_attributeset.xsl	Stylesheet that contains all xsl:attribute definitions.
dbtofo_const.xsl	Stylesheet that contains global variables.
dbtofo_content.xsl	Stylesheet that manipulates main DocBook elements.
dbtofo_global.xsl	Stylesheet that defines data/parameter dependent global variables.
dbtofo_index.xsl	Stylesheet that passes indexterm element to the library and make index page from results.
dbtofo_indexterm.xsl	Stylesheet that checks indexterm integrity and generates fo:wrapper element and XSL1.1 indexkey property.
dbtofo_main.xsl	Stylesheet that contains main control templates.
dbtofo_param.xsl	Stylesheet that contains parameters.
dbtofo_en.xsl dbtofo_ja.xsl dbtofo_ko.xsl dbtofo_zhcn.xsl dbtofo_zhcn_sortaspinyin.xsl dbtofo_zhtw.xsl	Language wrapper stylesheet that imports dbtofo.xsl.
xslt_common.xsll	Stylesheet that supply XSLT processor information. (XSLT1.0 only)

The most important stylesheet files are dbtofo_indexterm.xsl and dbtofo_index.xsl. The core index processing algorithms is integrated into these two stylesheets. You can make modifications to this stylesheet or import them into your stylesheets.

5 Getting Started With DITA Open Toolkit PDF5-ML plug-in

5.1 Downloading DITA Open Toolkit

You can download DITA Open Toolkit from following URL.

http://www.dita-ot.org/

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

5.2 Updating DITA Open Toolkit batch file

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

```
REM AH Formatter home and setting file
set AHF_DIR=C:\Program Files\Antenna House\AHFormatterV62
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 dita folder. Please copy it to the [DITA-OT] folder.

5.3 Installing I18n Index Library Plugin

1. Copy `dita\com.antennahouse.i18n_index.2.3` folder to `[DITA-OT]\plugins` folder.
2. Copy `dita\run_en.bat, run_ja.bat` files to `[DITA-OT]` folder.

These batch files explicitly set -Duse.i18n.index.lib=yes in the command-line. This is needed to invoke PDF5-ML with I18n Index Library.

```
ant -l out\sample_en.log -Dargs.input=samples/index-data/sample_en/sample_en.ditamap
-Doutput.draft.comment=yes -Doutput.required.cleanup=yes -Dtranstype=pdf5.ml -
Duse.i18n.index.lib=yes
```

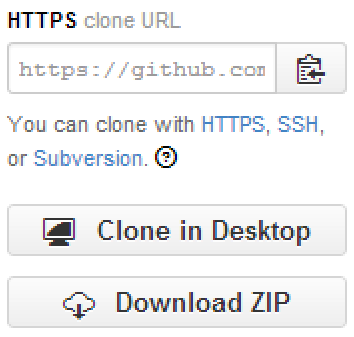
5.4 Installing PDF5-ML Plugin

Follow the next instructions.

1. Download PDF5-ML plug-in from GitHub https://github.com/AntennaHouse/pdf5-ml.

   You can get ZIP file named pdf5-ml-master.zip by clicking the "Download ZIP" button located on the right side of this page.

[Fig.5] Downloading PDF5-ML ZIP file



2. Unzip pdf5-ml-master.zip.
3. Copy `com.antennahouse.pdf5.ml` folder to `[DITA-OT]\plugins` folder.
4. Copy `samples\sample_en, sample_ja` folder to `[DITA-OT]\samples` folder.
5. Make `[DITA-OT]\out` folder.

NOTE: *Originally PDF5-ML plug-in is configured to work without I18n Index Library plug-in. To work with I18n Index Library you must set -Duse.i18n.index.lib=yes in command-line. Do not use batch file run_en.bat, run_ja.bat bundled with PDF5-ML plug-in ZIP file. These files do not contains this command-*

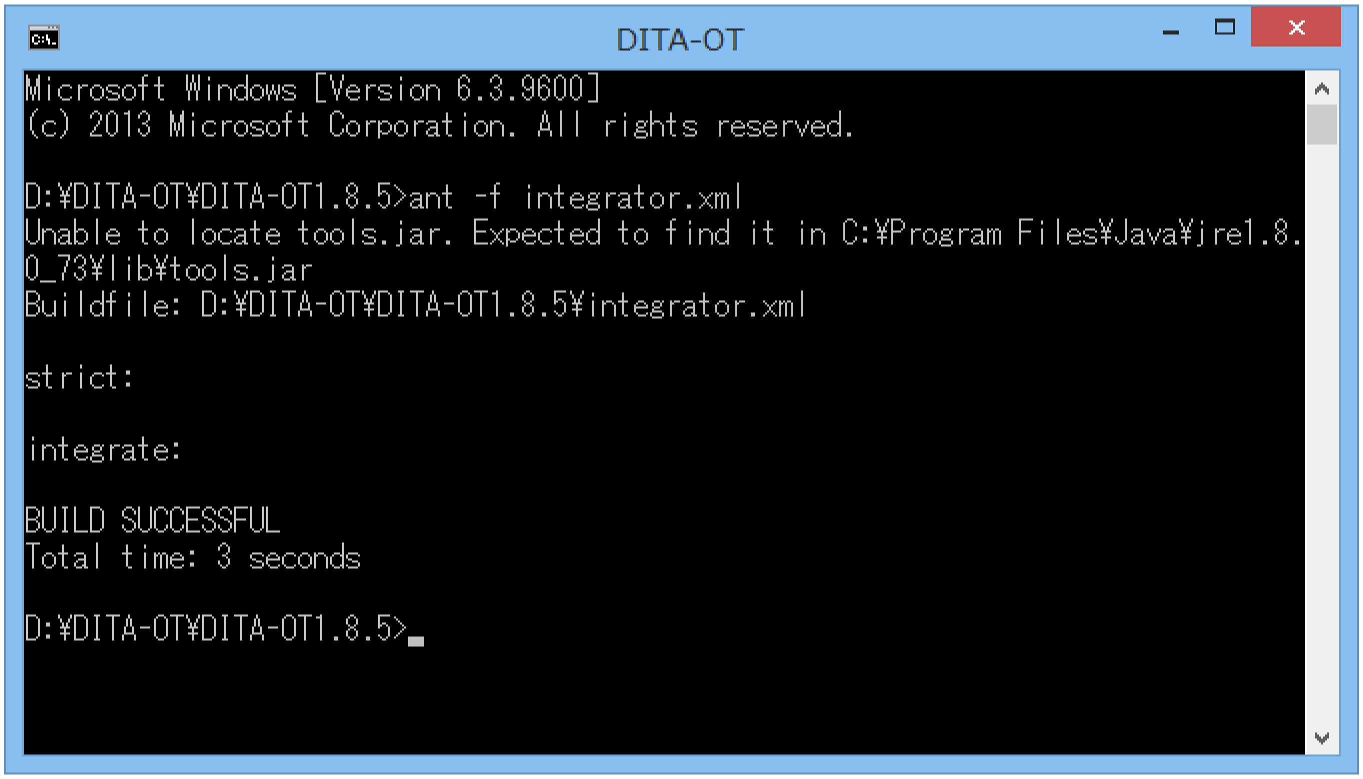
line setting.

5.5 Integrating plug-in into DITA-OT

1. From Explorer click `[DITA-OT]\startcmd.bat` file. The command window titled "DITA-OT" opens.
2. From command window enter the following command. This command integrates I18n Index Library and PDF5-ML plug-ins into DITA Open Toolkit.

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

   [Fig.6] Integrating plug-ins into DITA Open Toolkit

   ```
   Microsoft Windows [Version 6.3.9600]
   (c) 2013 Microsoft Corporation. All rights reserved.

   D:¥DITA-OT¥DITA-OT1.8.5>ant -f integrator.xml
   Unable to locate tools.jar. Expected to find it in C:¥Program Files¥Java¥jre1.8.
   0_73¥lib¥tools.jar
   Buildfile: D:¥DITA-OT¥DITA-OT1.8.5¥integrator.xml

   strict:

   integrate:

   BUILD SUCCESSFUL
   Total time: 3 seconds

   D:¥DITA-OT¥DITA-OT1.8.5>_
   ```

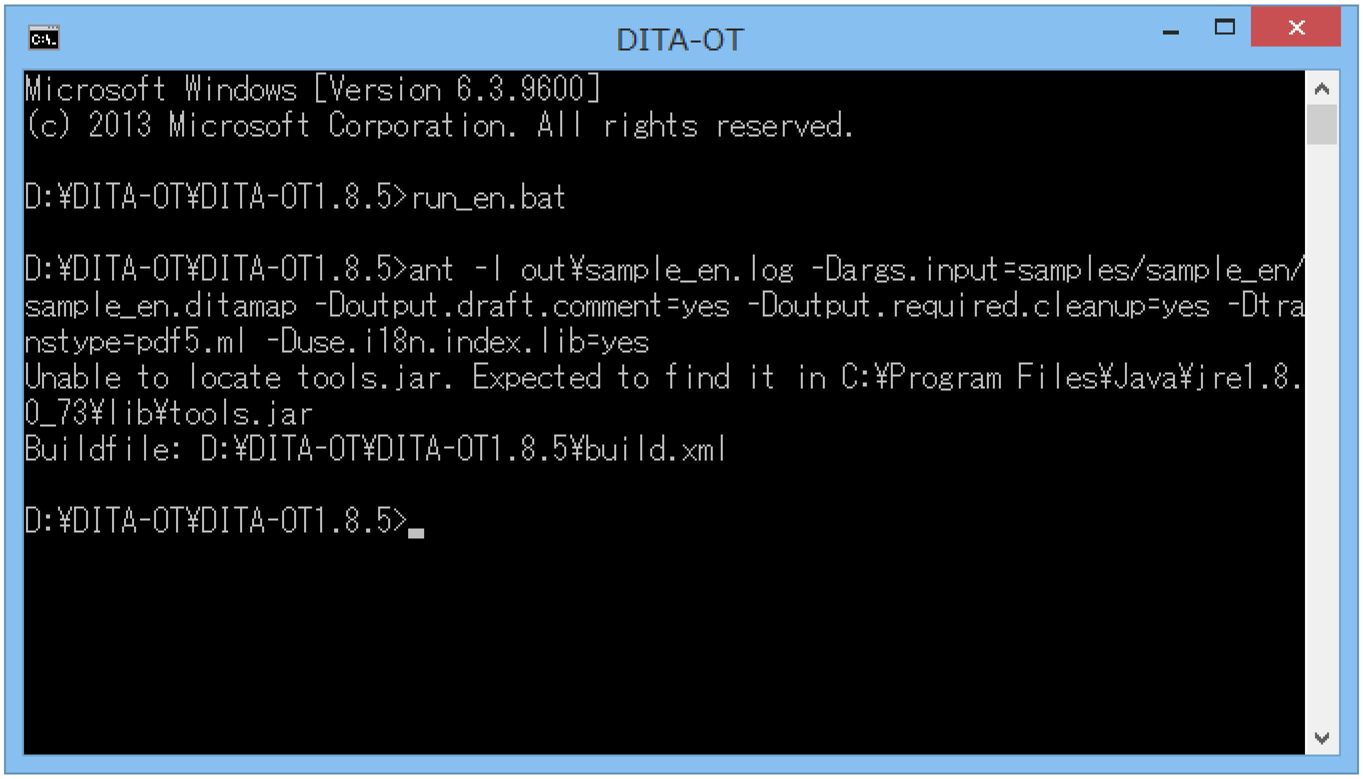
3. Close the command window.

5.6 Testing PDF Output

1. From Explorer click `[DITA-OT]\startcmd.bat` file. The command window titled "DITA-OT" opens.
2. You can test following batch files from this window.
   - run_en.bat
   - run_ja.bat

   These batch files have no parameter.

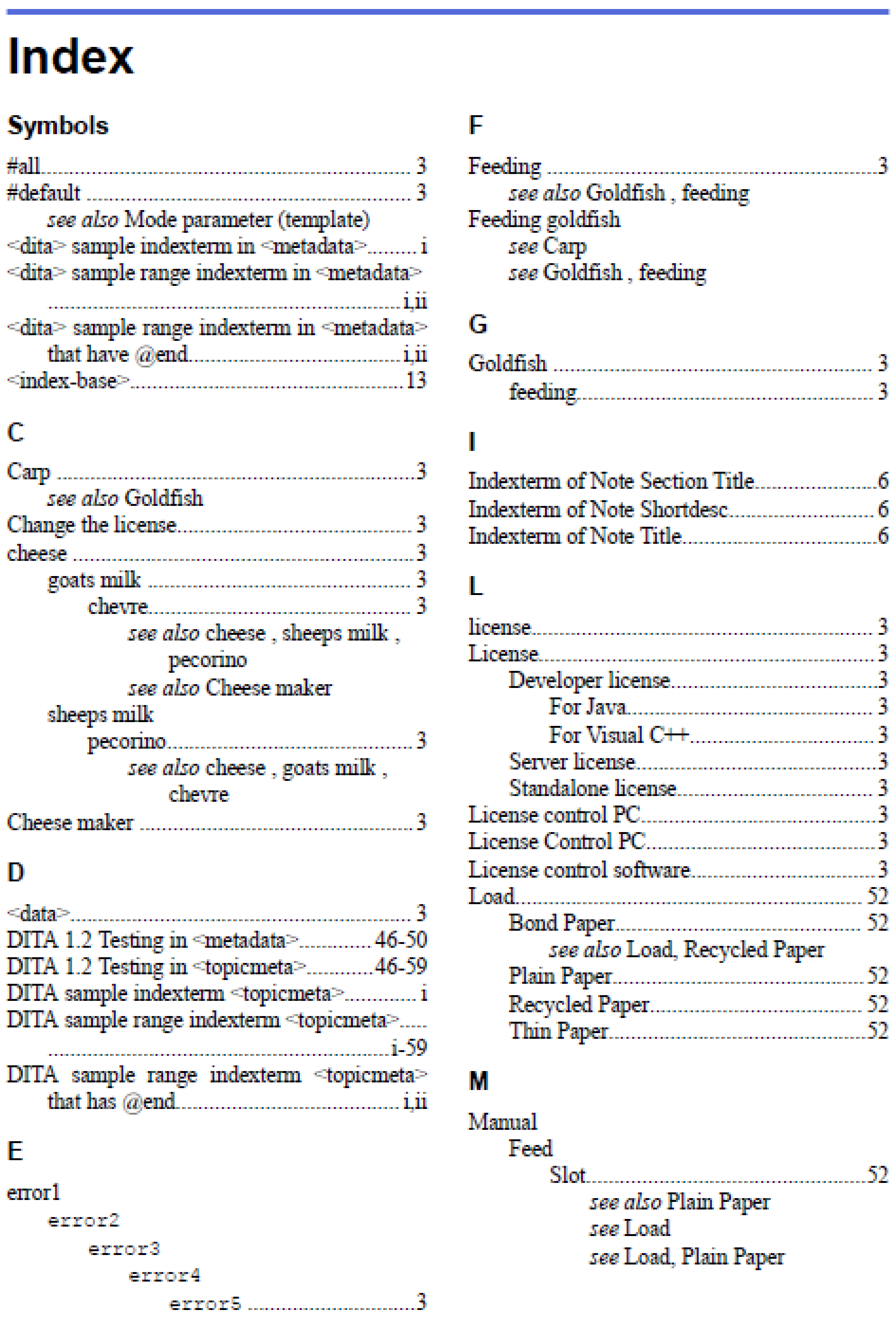
   [Fig.7] Example of running a batch file in DITA Open Toolkit



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

D:¥DITA-OT¥DITA-OT1.8.5>run_en.bat

D:¥DITA-OT¥DITA-OT1.8.5>ant -l out¥sample_en.log -Dargs.input=samples/sample_en/
sample_en.ditamap -Doutput.draft.comment=yes -Doutput.required.cleanup=yes -Dtra
nstype=pdf5.ml -Duse.i18n.index.lib=yes
Unable to locate tools.jar. Expected to find it in C:¥Program Files¥Java¥jre1.8.
0_73¥lib¥tools.jar
Buildfile: D:¥DITA-OT¥DITA-OT1.8.5¥build.xml

D:¥DITA-OT¥DITA-OT1.8.5>_
```

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

[Fig.8] English PDF sample of DITA Open Toolkit



Index

Symbols

#all ... 3
#default ... 3
  see also Mode parameter (template)
<dita> sample indexterm in <metadata> ... i
<dita> sample range indexterm in <metadata> ... i,ii
<dita> sample range indexterm in <metadata> that have @end ... i,ii
<index-base> ... 13

C

Carp ... 3
  see also Goldfish
Change the license ... 3
cheese ... 3
  goats milk ... 3
    chevre ... 3
      see also cheese , sheeps milk , pecorino
      see also Cheese maker
  sheeps milk
    pecorino ... 3
      see also cheese , goats milk , chevre
Cheese maker ... 3

D

<data> ... 3
DITA 1.2 Testing in <metadata> ... 46-50
DITA 1.2 Testing in <topicmeta> ... 46-59
DITA sample indexterm <topicmeta> ... i
DITA sample range indexterm <topicmeta> ... i-59
DITA sample range indexterm <topicmeta> that has @end ... i,ii

E

error1
  `error2`
    `error3`
      `error4`
        `error5` ... 3

F

Feeding ... 3
  see also Goldfish , feeding
Feeding goldfish
  see Carp
  see Goldfish , feeding

G

Goldfish ... 3
  feeding ... 3

I

Indexterm of Note Section Title ... 6
Indexterm of Note Shortdesc ... 6
Indexterm of Note Title ... 6

L

license ... 3
License ... 3
  Developer license ... 3
    For Java ... 3
    For Visual C++ ... 3
  Server license ... 3
  Standalone license ... 3
License control PC ... 3
License Control PC ... 3
License control software ... 3
Load ... 52
  Bond Paper ... 52
    see also Load, Recycled Paper
  Plain Paper ... 52
  Recycled Paper ... 52
  Thin Paper ... 52

M

Manual
  Feed
    Slot ... 52
      see also Plain Paper
      see Load
      see Load, Plain Paper

5.7 Test Data Features

By running the batch files, you can confirm the following features:

Batch File	DITA Map File	PDF File, Log File	Feature	Used Font
			DITA 1.3 and XSL 1.1 features; • Multiple level nested <indexterm>. • Multiple <index-see> and	

run_en.bat	sample_en.ditamap	sample_en.pdf sample_en.log	<index-see-also> feature. • Change index position using <index-sortas> element. (Sample: <data/>) • Range indexterm/@start,@end. (Range index is displayed using hyphen: "1-3, 5-7") • sample_en.pdf contains various DITA 1.1 elements and attributes.	Times New Roman, Arial, Courier New
run_ja.bat	sample_ja.ditamap	sample_ja.pdf sample_ja.log	Japanese index feature; This is same as DocBook.	MS Mincho, MS Gothic

NOTE: *Above fonts in the table are all bundled in Windows. If you want to use this plugin in other operating environment, you must change fonts for each language. For customization, refer to pdf5-ml_manual.pdf bundled in pdf5-ml-master.zip for details.*

5.8 Customizing Stylesheet

In this plugin the stylesheets can be found `[DITA-OT]\pluins\com.antennahouse.pdf5.ml\xsl` folder. As DITA Open Toolkit bundles Saxon-B 9.1 as XSLT processor, these stylesheets are written in XSLT 2.0. If you want to customize this stylesheets, refer to pdf5-ml_manual.pdf bundled in pdf5-ml-master.zip.

6 Library Interface

6.1 I18n Index Library .jar file

6.1.1 DocBook

Sample DocBook stylesheet uses following .jar file.

```
docbook\i18n_index\ah_i18n_index.jar
docbook\i18n_index\i18n_support.jar
docbook\i18n_index\i18n_support\lib\icu4j-56_1.jar
```

6.1.2 I18n Index Library plug-in

I18n Index Library plug-in has following ant build file configurations to use index sorting from DITA-OT plug-in stylesheet.

```
<path id="i18n_index.class.path">
    <fileset
dir="${com.antennahouse.i18n_index.plugin.lib}${file.separator}i18n_support${file.separator}lib">
        <include name="*.jar"/>
    </fileset>
    <fileset dir="${com.antennahouse.i18n_index.plugin.lib}">
        <include name="*.jar"/>
    </fileset>
</path>
```

The property `com.antennahouse.i18n_index.plugin.lib` is exposed from DITA-OT main build.xml. If you want to use I18n Index Library plug-in in DITA-OT, include this path to your XSLT step classpath.

6.2 Public method for indexterm sorting

The library ah_i18n_index.jar has the following static method for the stylesheet (XSLT processor).

XSLT processor	Class	Method
Xalan-J 2.7.1/2.7.2	jp.co.antenna.ah_i18n_index.IndexSortXalan2	public static org.w3c.dom.DocumentFragment indexSortXalan2(String lang, org.w3c.dom.NodeList nodeList)
		public static org.w3c.dom.DocumentFragment indexSortXalan2(String lang, org.w3c.dom.NodeList nodeList, String assumeSortasPinyin)
Saxon 6.5.5	jp.co.antenna.ah_i18n_index.IndexSortSaxon6	public static org.w3c.dom.NodeList indexSortSaxon6(String lang, com.icl.saxon.expr.FragmentValue indextermInfo)
		public static org.w3c.dom.NodeList indexSortSaxon6(String lang, com.icl.saxon.expr.FragmentValue indextermInfo, String assumeSortasPinyin)
Saxon-B 9.1, Saxon PE/EE 9.2 or later	jp.co.antenna.ah_i18n_index.IndexSortSaxon9	public static org.w3c.dom.NodeList indexSortSaxon9(String lang, org.w3c.dom.Node indextermInfo)
		public static org.w3c.dom.NodeList indexSortSaxon9(String lang, org.w3c.dom.Node indextermInfo, String assumeSortasPinyin)

The first parameter is a language code defined in RFC3066. It should be the main language of the input document.

The second parameter differs by XSLT processor. It represents the same temporary document (Result Tree Fragments in XSLT 1.0) created from the indexterm element. The content has the following element structure.

```
    <!-- Indexterm element in source document -->
    <indexterm>
        <primary sortas="element">&amp;lt;element/&amp;gt;</primary>
    </indexterm>
    <indexterm>
        <primary>cheese</primary>
        <secondary>sheeps milk</secondary>
        <tertiary>pecorino</tertiary>
    </indexterm>

         ↓

    <!-- Temporary document passed from stylesheet to library -->
    <index-data indexkey="&amp;lt;element/&amp;gt" level="1" nestedindexterm="0"
significance="normal">
        <indexterm sortas="element">&amp;lt;element/&amp;gt</indexterm>
    </index-data>
    <index-data indexkey="cheese:sheeps milk:pecorino" level="3" nestedindexterm="0"
significance="normal">
```

```
        <indexterm>cheese</indexterm>
        <indexterm>sheeps milk</indexterm>
        <indexterm>pecorino</indexterm>
    </index-data>

         ↓

    <!-- Temporary document passed from library to stylesheet -->
    <index-data group-key="C" group-label="C" group-sort-key="C" id="d0e26"
indexkey="cheese:sheeps milk:pecorino" level="3" nestedindexterm="0" significance="normal">
        <indexterm>cheese</indexterm>
        <indexterm>sheeps milk</indexterm>
        <indexterm>pecorino</indexterm>
    </index-data>
    <index-data group-key="E" group-label="E" group-sort-key="E" id="d0e20"
indexkey="&lt;element/&gt;" level="1" nestedindexterm="0" significance="normal">
        <indexterm sortas="element">&lt;element/&gt;</indexterm>
    </index-data>
```

The third parameter is a string that indicates to use the sortas attribute as pinyin reading. This parameter value should be "true" or "false". The default value is "false" and this parameter can be omitted.

NOTE: *Refer to the JavaDoc in the javadoc folder or db2fo_index.xsl stylesheet source file for details.*

6.3 Note when using sortaspinyin="true"

In this mode, the sortas attribute is the assumed pinyin reading. So you can correct the irregularly-positioned indexterm like following.

```
<!-- Illegaly positioned indexterm in index page (粘贴 belongs N group)-->
<indexterm>
    <primary>粘贴</primary>
</indexterm>

     ↓

<!-- Corrected indexterm (粘贴 belongs Z group) -->
<indexterm>
    <primary sortas="zhan1 tie1">粘贴</primary>
</indexterm>
```

But there are other uses for sortas. For example, you might want to place "<element/>" indexterm to E group or you might want to treat "β测试" (β testing) as "测试". In this case, please modify the source document as follows.

```
<!-- sortaspinyin="false" mode -->
<indexterm>
    <primary sortas="element">&amp;lt;element/&amp;gt;</primary>
</indexterm>
<indexterm>
    <primary sortas="测试">β测试</primary>
</indexterm>

     ↓

<!-- sortaspinyin="true" mode ("Δ" means space) -->
<indexterm>
    <primary sortas="eΔlΔeΔmΔeΔnΔt">&amp;lt;element/&amp;gt;</primary>
</indexterm>
<indexterm>
    <primary sortas="&#xFFFD;Δce4Δshi4">β测试</primary>
</indexterm>
```

Because the sortas attribute is assumed as pinyin readings, Hanzi characters must be specified as pinyin and Non-Hanzi characters can be used as is. **However in both cases, pinyin or Non-Hanzi characters must both be separated by a space.** And in the latter example, you must insert U+FFFD (REPLACEMENT CHARACTER) instead of "β" character. This is because, the "β" character is not included in the sortas attribute. U+FFFD is a placeholder of such character.

6.4 Public method for general element sorting

The library ah_i18n_index.jar has the following static method to sort general elements for the stylesheet

(XSLT processor).

XSLT processor	Class	Method
Xalan-J 2.7.1/2.7.2	jp.co.antenna.ah_i18n_generalsort.GeneralSortXalan2	static org.w3c.dom.DocumentFragment generalSortXalan2(java.lang.String lang, org.w3c.dom.NodeList nodeList)
		static org.w3c.dom.DocumentFragment generalSortXalan2(java.lang.String lang, org.w3c.dom.NodeList nodeList, java.lang.String assumeSortasPinyin)
Saxon 6.5.5	jp.co.antenna.ah_i18n_generalsort.GeneralSortSaxon6	static org.w3c.dom.NodeList generalSortSaxon6(java.lang.String lang, com.icl.saxon.expr.FragmentValue generalInfo)
		static org.w3c.dom.NodeList generalSortSaxon6(java.lang.String lang, com.icl.saxon.expr.FragmentValue generalInfo, java.lang.String assumeSortasPinyin)
Saxon-B 9.1, Saxon PE/EE 9.2 or later	jp.co.antenna.ah_i18n_generalsort.GeneralSortSaxon9	static org.w3c.dom.NodeList generalSortSaxon9(java.lang.String lang, org.w3c.dom.Node generalInfo)
		static org.w3c.dom.NodeList generalSortSaxon9(java.lang.String lang, org.w3c.dom.Node generalInfo, java.lang.String assumeSortasPinyin)

The first parameter is a language code defined in RFC3066. It should be the main language of the input document.

The second parameter differs by XSLT processor. It represents the same temporary document (Result Tree Fragments in XSLT 1.0) created from the target sorting element. The content has the following element structure.

```
    <!-- topicref in source document -->
    <topicref href="xmlelement.dita" format="dita">
        <topicmeta><navtitle>&gt;xmlelement&lt;<sort-as>xmlelement</sort-
 as></navtitle></topicmeta>
    </topicref>
    <topicref href="relaxng.dita" format="dita">
        <topicmeta><navtitle>RELAX NG</navtitle></topicmeta>
    </topicref>
    <topicref href="dita1.3.dita" format="dita">
        <topicmeta><navtitle>DITA 1.3</navtitle></topicmeta>
    </topicref>
```

```
          ↓

    <!-- Temporary document passed from stylesheet to library.
         Specify sort key by @sort-key and alternative sort key as @sort-as.
     -->
    <topicref href="xmlelement.dita" format="dita" sort-key="&gt;xmlelement&lt;" sort-
as="xmlelement">
       <topicmeta>
          <navtitle>&gt;xmlelement&lt;<sort-as>xmlelement</sort-as>
          </navtitle>
       </topicmeta>
    </topicref>
    <topicref href="relaxng.dita" format="dita" sort-key="RELAX NG">
       <topicmeta>
          <navtitle>RELAX NG</navtitle>
       </topicmeta>
    </topicref>
    <topicref href="dita1.3.dita" format="dita" sort-key="DITA 1.3">
       <topicmeta>
          <navtitle>DITA 1.3</navtitle>
       </topicmeta>
    </topicref>

          ↓

    <!-- Sorted temporary document passed from library to stylesheet.
         You can perform grouping sort results by <xsl:for-each-group> using @group-key
attribute as group key.
     -->
    <topicref format="dita" group-key="D" group-label="D" group-sort-key="D" href="dita1.3.dita"
sort-key="DITA 1.3">
       <topicmeta>
          <navtitle>DITA 1.3</navtitle>
       </topicmeta>
    </topicref>
    <topicref format="dita" group-key="R" group-label="R" group-sort-key="R" href="relaxng.dita"
sort-key="RELAX NG">
       <topicmeta>
          <navtitle>RELAX NG</navtitle>
       </topicmeta>
    </topicref>
    <topicref format="dita" group-key="X" group-label="X" group-sort-key="X"
href="xmlelement.dita" sort-as="xmlelement" sort-key="&gt;xmlelement&lt;">
       <topicmeta>
          <navtitle>&gt;xmlelement&lt;<sort-as>xmlelement</sort-as>
          </navtitle>
       </topicmeta>
    </topicref>
```

The third parameter is a string that indicates to use the sortas attribute as pinyin reading. This parameter value should be "true" or "false". The default value is "false" and this parameter can be omitted.

NOTE: *Refer to the JavaDoc in the javadoc folder for details.*

7 Other Resources

- This release contains I18n Support Library source codes and documents in **i18n_support_lgpl.zip**. To see the source codes or the documents, unzip **i18n_support_lgpl.zip**. The documents are remained without modification from original version in the docs folder.
- If you want to change index configurations, you must change i18n_index/config/botb_index_rules/botb_index_rules.xml. For more information read the documents mentioned above (docs/isogen_i18n_user_guide.html in i18n_support_lgpl.zip) and the comments in the botb_index_rules.xml file.
- If you want to customize index sorting algorithms or use another Java based XSLT processor, you can customize and rebuild ah_i18n_index.jar by using **ah_i18n_index_nolgpl.zip**. This archive contains the source code of I18n Index Library.

< END OF DOCUMENT >