

Taking DITA for a Loop

 *Automating XLIFF Conversions*

infotexture

Information Architecture & Content Strategy

Roger W. Fienhold Sheen

Agenda

Darwin Information Typing Architecture

DITA Open Toolkit

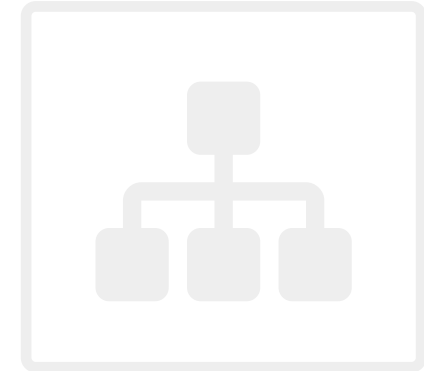
DITA-XLIFF Roundtrip Plugins

Installing the Plugins

Exporting XLIFF from DITA

Generating DITA Files from XLIFF

Automating Conversions



Darwin Information Typing Architecture



“An XML architecture for designing, writing, managing, and publishing information.”

— dita.xml.org

Originally developed by IBM for internal use, DITA has since become an open standard published by OASIS, the *Organization for the Advancement of Structured Information Standards* — like XLIFF.

- future-proof & interoperable, widely adopted & rapidly gaining traction
- well-supported via open-source toolkit & many commercial tools
- encourages good documentation practices
- provides built-in support for crucial use cases, including
 - modular content re-use
 - multiple output formats
 - localization

DITA Open Toolkit



“The DITA Open Toolkit is a set of Java-based, open source tools that provide processing for DITA maps and topic content.”

The **DITA Open Toolkit** is primarily a publishing tool used to convert (“transform”) DITA content into various output formats, including:

- **PDF** — via XSL-FO renderers (*Apache FOP, RenderX, Antenna House*)
- **XHTML** — Plain HTML with class values to permit styling via CSS
- **TocJS** – HTML output with JavaScript-based navigation frames
- **HTML Help** — Microsoft Compiled HTML Help output (.chm)
- **Eclipse Help, Java Help** — HTML with navigation & index
- **OpenDocument** — Based on the ODF standard, for Open Office, etc.
- **Rich Text Format** — Basic content supported, but not complex markup

The toolkit’s plug-in mechanism allows new transformation types to be added.

DITA-XLIFF Roundtrip Plugins



The **DITA-XLIFF Roundtrip** plugins for the DITA Open Toolkit convert DITA content to *XML Localization Interchange File Format* and create DITA topics from XLIFF files.

The plugins are provided by **OASIS XLIFF TC** chair Bryan Schnabel.

No CMS Required

- Users of commercial content management systems may rely on the export/import mechanisms provided by the CMS vendor.
- How to extract necessary content for localization and re-import translated material in file-system-based projects on limited budgets?
- The plugins provide a viable open-source mechanism for packaging an entire project of DITA content as a single XLIFF file that can be easily exchanged with localization service providers and translation vendors.

Benefits

- **Content owners retain control of the conversion process**
Keep conversion close to creation — content owners can use the plugins to extract the localized content from the XLIFF file and automatically generate the necessary DITA files in a separate source tree for the target language.
- **Minimize complexity** — errors are inevitable when hundreds or thousands of individual files are converted and exchanged with the vendor
- **Avoid vendor lock-in** — since the vendor doesn't own the “black box” conversion process, it's easier to switch vendors if necessary
- **Ensure consistency** — via automated (*and thus reproducible*) processes
- **Save time & money** — *no more “file management” fees!*

Package Contents

The distribution package includes four separate plugins:

- The `xliff` plugins extend the toolkit with new transformation types that can be used to transform an entire DITA project to a single XLIFF file
- The `ditafromxliff` plugins read an XLIFF file and re-generate the entire hierarchy of translated DITA topics from the translated XLIFF content

Two versions of each conversion plugin are provided:

- The `v12` variants generate or convert from the **XLIFF version 1.2** format
- The `v2` plugins create and read **XLIFF version 2.0**

Installing the Plugins

In DITA Open Toolkit, install the plugins as follows:

1. Download the latest distribution package for the plugins from sourceforge.net/projects/ditaxliff/files/latest/download.
2. Unzip the package to the `plugins` directory of your DITA-OT installation.
3. From the root directory of your DITA-OT installation, run the start command for your operating system (either `startcmd.bat` or `startcmd.sh`).
4. In the `startcmd` environment, run the following command to integrate the new plug-ins into your toolkit installation:

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

Before You Begin...

Contact your vendor to find out which version of XLIFF they support.



Exporting XLIFF from DITA

1. Copy the source files for your project to a new subfolder in the `plugins/xliff-v*/samples` folder (i.e. `xliff-v*/samples/my-project`).
2. Edit the value of the `args.input` property on line 27 of the Ant build script (`xliff-v*/build_dita2xliff.xml`) to point to your map.



Change the line that looks like this:

```
<property name="args.input" value="samples/guitars/sample.ditamap"/>
```

to something like this:

```
<property name="args.input" value="samples/my-project/my-map.ditamap"/>
```

3. In the `xliff-v*` directory, run the Ant build script.

This script generates the XLIFF file for your DITA topics:

```
ant -f build_dita2xliff.xml
```

```
xliff-v2 — bash — infotexture

[fof] INFO: Rendered page #5.
[fof] May 30, 2015 6:58:19 PM org.apache.fof.events.LoggingEventListener processEvent
[fof] INFO: Rendered page #6.
[fof] May 30, 2015 6:58:19 PM org.apache.fof.events.LoggingEventListener processEvent
[fof] INFO: Rendered page #7.
[fof] May 30, 2015 6:58:19 PM org.apache.fof.events.LoggingEventListener processEvent
[fof] INFO: Rendered page #8.
[fof] May 30, 2015 6:58:19 PM org.apache.fof.events.LoggingEventListener processEvent
[fof] INFO: Rendered page #9.
[fof] May 30, 2015 6:58:19 PM org.apache.fof.events.LoggingEventListener processEvent
[fof] INFO: Rendered page #10.
[fof] May 30, 2015 6:58:19 PM org.apache.fof.events.LoggingEventListener processEvent
[fof] INFO: Rendered page #11.
[fof] May 30, 2015 6:58:19 PM org.apache.fof.events.LoggingEventListener processEvent
[fof] WARNING: Bookmarks: Unresolved ID reference "ID_INDEX_00-0F-EA-48-0D-4D" found.
[fof] /Users/rofish/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/plugins/xliff-v2/temp/topic.fo -> /Users/rofish/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/plugins/xliff-v2/out/samples/xliff/sample.pdf

transform.fo2pdf:
delete.fo2pdf.topic.fo:
transform.topic2pdf:
publish.map.pdf:
topic2pdf2:
d2xSL:
[xs1t] Processing /Users/rofish/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/plugins/xliff-v2/temp/stage1.xml to /Users/rofish/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/plugins/xliff-v2/out/samples/xliff/d_x.xml
[xs1t] Loading stylesheet /Users/rofish/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/plugins/xliff-v2/xsl/d2x.xsl

dita2xliff:
clean-temp:
dita2xliff:

BUILD SUCCESSFUL
Total time: 10 seconds

rofish @ infotexture-macbook-pro ~/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/plugins/xliff-v2
$
```

RESULT: The `d_x.xml` file is created in the `xliff-v*/out/samples/xliff` folder.

This is a valid XLIFF file that has all your maps, topics, and required structure to translate and reassemble your DITA project.

Translation via Localization Vendor

1. You provide the generated XLIFF file to your localization vendor.



magic happens...

— a hitherto unknown version of your content emerges...



2. Your localization vendor returns the translated XLIFF file to you.



Generating DITA Files from XLIFF

1. Change the name of the translated file to `d_x_translated.xml`.
2. Place it in the `plugins/ditafromxliff-v*/in` directory.
3. In `plugins/ditafromxliff-v*`, run the Ant build script.

This script generates the DITA topics from the XLIFF file:

```
ant -f build_ditafromxliff.xml
```

4. Switch to the `ditafromxliff-v*/out/samples/ditafromxliff` directory.

RESULT: The source tree for the target language appears in the `translated` subfolder.



```
ditafromxliff-v2 — bash — infotexture
mappull:
chunk-check:
chunk:
maplink-check:
maplink:
move-links-check:
move-links:
topicpull-check:
topicpull:
[topicpull] Transforming into /Users/rofish/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/plugins/ditafromxliff-v2/temp
[topicpull] Processing /Users/rofish/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/plugins/ditafromxliff-v2/temp/in/d_x_translated.xml to /Use
rs/rofish/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/plugins/ditafromxliff-v2/temp/in/d_x_translated.xml.pull
[topicpull] Loading stylesheet /Users/rofish/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/xsl/preprocess/topicpull.xsl
[move] Moving 1 file to /Users/rofish/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/plugins/ditafromxliff-v2/temp
flag-module-check:
flag-module:
preprocess:
map2pdf2:
x2dXSL:
dita2ditafromxliff:
clean-temp:
dita2ditafromxliff:
BUILD SUCCESSFUL
Total time: 3 seconds
rofish @ infotexture-macbook-pro ~/Documents/Work/infotexture/Projects/DITA/DITA-0T1.8.5/plugins/ditafromxliff-v2
$
```

Automating Conversions

Continuous Integration servers communicate with a version control system to monitor source files and perform tasks when files change.



If your developers use continuous integration tools to run automated tests and build software binaries whenever they update their code, you may be able to use the same solution to build DITA deliverables and convert to/from XLIFF.

Jenkins, one of the most popular CI solutions, allows you to define jobs that combine various settings, including:

- access credentials and branches of the source code repository
- conditions or events that trigger a build, and
- actions to be performed when the conditions are fulfilled
(*build script & post-build actions, e-mail notifications, file transfers, etc.*).

```
ant -f build_dita2xliff.xml
```

Jenkins Dashboard

The Jenkins dashboard includes an overview of jobs with information on the last build for each job, and a “weather report” icon that represents the aggregated status (*stability*) of recent builds:

Jenkins > Documentation > ENABLE AUTO REFRESH

Welcome to the Adyton Systems Continuous Integration Server

Alle Clumsy Core **Documentation** IT Test Web

S	W	Name ↓	Last Success	Last Failure	Last Duration
		ci-documentation-master	1 mo 8 days - #6	N/A	3 min 12 sec
		ci-documentation-release-4.4	4 mo 24 days - #21	5 mo 15 days - #3	4 min 2 sec
		Documentation (Core)	3 hr 25 min - #804	2 mo 4 days - #690	23 sec
		Documentation (Manual)	1 day 2 hr - #39	1 mo 21 days - #21	1 min 17 sec
		Documentation (Web)	7 mo 25 days - #80	8 mo 12 days - #50	37 sec
		W Description	%		
		Build stability: No recent builds failed.	100		N/A

Icon: [S](#) [M](#) [L](#)

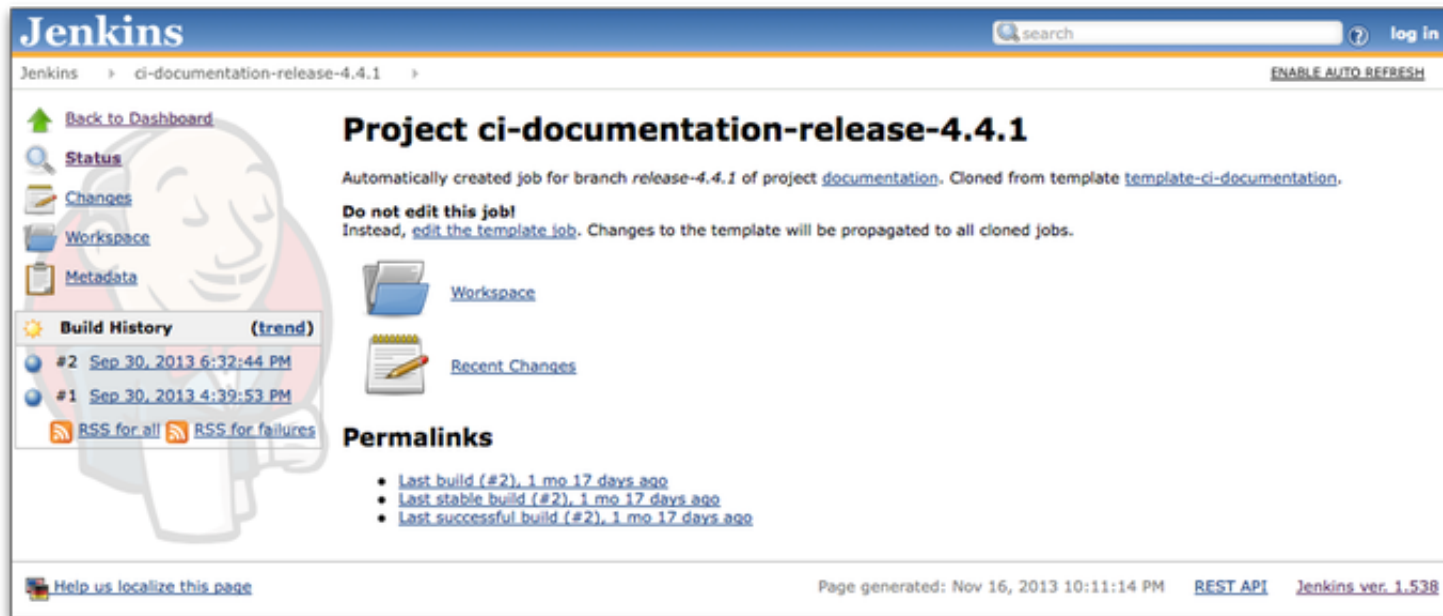
[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

[Help us localize this page](#) Page generated: Jan 31, 2014 10:26:38 PM [REST API](#) [Jenkins ver. 1.542](#)

Sample Jenkins dashboard

Jenkins Job View

The dashboard links to dedicated pages for each job, with additional information on the build history, links to the workspace with the job output (*build results*), and recent changes (*the commit log from the version control system*).



The screenshot shows the Jenkins web interface for a specific job. The header includes the Jenkins logo, a search bar, and a 'log in' link. The breadcrumb trail shows 'Jenkins > ci-documentation-release-4.4.1'. A sidebar on the left contains navigation links: 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Metadata', and 'Build History (trend)'. The 'Build History' section lists two builds: '#2 Sep 30, 2013 6:32:44 PM' and '#1 Sep 30, 2013 4:39:53 PM', with RSS feeds for all builds and failures. The main content area is titled 'Project ci-documentation-release-4.4.1' and contains a warning: 'Do not edit this job! Instead, edit the template job.' Below this are links for 'Workspace' and 'Recent Changes'. A 'Permalinks' section lists: 'Last build (#2), 1 mo 17 days ago', 'Last stable build (#2), 1 mo 17 days ago', and 'Last successful build (#2), 1 mo 17 days ago'. The footer includes a localization link, the page generation timestamp 'Nov 16, 2013 10:11:14 PM', and links for 'REST API' and 'Jenkins ver. 1.538'.

Sample Jenkins job view

Resources

- [DITA Feature Article: “Using XLIFF to Translate DITA Projects”](#)



Updates

For updates, comments and code samples, visit <http://infotexture.net>.

Contact

✉ E-mail info@infotexture.net

in LinkedIn [@infotexture](#)

 GitHub [@infotexture](#)

 Twitter [@infotexture](#)

 **Wire** Roger Sheen