An OASIS DITA Adoption Technical Committee Publication

DITA 1.2 Feature Article: Using XLIFF to Translate DITA Projects

Author: Rodolfo Raya, Bryan Schnabel, and JoAnn Hackos
On behalf of the DITA Adoption Technical Committee

Date: 28 Jan 2013

Table of Contents

OASIS (Organization for the Advancement of Structured Information Standards) is a not-for-profit, international consortium that drives the development, convergence, and adoption of e-business standards. Members themselves set the OASIS technical agenda, using a lightweight, open process expressly designed to promote industry consensus and unite disparate efforts. The consortium produces open standards for Web services, security, e-business, and standardization efforts in the public sector and for application-specific markets. OASIS was founded in 1993. More information can be found on the OASIS website at http://www.oasis-open.org.

The OASIS DITA Adoption Technical Committee members collaborate to provide expertise and resources to educate the marketplace on the value of the DITA OASIS standard. By raising awareness of the benefits offered by DITA, the DITA Adoption Technical Committee expects the demand for, and availability of, DITA conforming products and services to increase, resulting in a greater choice of tools and platforms and an expanded DITA community of users, suppliers, and consultants.

DISCLAIMER: All examples presented in this article were produced using one or more tools chosen at the author's discretion and in no way reflect endorsement of the tools by the OASIS DITA Adoption Technical Committee.

This white paper was produced and approved by the OASIS DITA Adoption Technical Committee as a Committee Draft. It has not been reviewed and/or approved by the OASIS membership at-large.

Document History

Revision	Date	Author	Summary
First Draft	29 August 2011	Hackos	Draft of the Committee Note -- Feature Article
Draft	7 October 2011	Hackos, Schnabel, Raya	Draft for committee vote
Draft	7 March 2012	Raya	Added workflow diagrams
Committee Approved Draft	21 May 2012	Hackos (Chair)	Adoption TC approved final draft

Using XLIFF to Translate DITA Projects

DITA promises cost reductions in translation thanks to content reuse. Content reuse can lead to translation cost reduction when proper planning is done. DITA topics are written once, updated once, and used in multiple deliverables. With careful planning, costs can indeed be lowered. However, cost reduction depends on the nature of the project; it is not possible to predict the level of cost reductions in advance.

Ideally, you should never pay twice for the translation of your content. The real world is not perfect, but with careful selection of tools and strategies, you can get excellent results.

If you use DITA topics in the different DITA maps that make up your projects, you can also reuse the translations of those topics.

XLIFF (XML Localization Interchange File Format) is an open standard published by OASIS (like DITA) that you can use in your project workflow to manage the content that needs to be translated.

The XLIFF vocabulary has a rich set of elements and attributes that permit XLIFF-supporting applications to

- store source and translated text
- store alternative or suggested translations extracted from a Translation Memory (TM) system or generated by a Machine Translation (MT) engine
- perform version control
- keep track of the different stages of the translation process
- perform word-count calculations

The XLIFF standard was first published by OASIS in 2002. Most modern translation environments currently support it. A Localization Service Provider (LSP) using up-to-date tools should be able to accept XLIFF files that you pre-process in-house.

You can translate DITA maps or individual topics using XLIFF as an intermediate format. When translating DITA maps, it is very important to use a tool that is aware of the different content-linking mechanisms offered by DITA. It is not necessary to resolve referenced content when translating individual topics, but you can still make good use of the translation reuse strategies described in this article.

Initial workflow

As a technical author, you write your DITA topics, prepare your maps, and publish source-language drafts as you would usually do. Refine your content as necessary, until you are satisfied with the published results. Once your project is ready for translation, convert your maps and topics to XLIFF format using a translation tool that supports DITA.

A translation tool that supports DITA will be able to

- resolve referenced content using *@href*, *@conref*, and other referencing attributes
- read the map(s) and resolve the topic references and generate matched-hierarchical XLIFF
- understand the standard “translate” attribute and the "xml:lang" attribute
- support your DITA specializations by
  - validating your DITA specializations by using, for example, OASIS XML catalogs or other open standards to invoke the appropriate DTD or Schema
  - providing a means to indicate which elements and attributes are translatable

The conversion to XLIFF will flatten the DITA map to a single file with all of the references resolved. The XLIFF file allows your Localization Service Provider (LSP) to work with one file rather than hundreds of files, simplifying administrative activities and reducing administrative costs.

The initial workflow proceeds as follows:

1. Begin with your DITA map in the source language (birds.ditamap).

```
<?xml version="1.0"?>
<!DOCTYPE map PUBLIC "-//OASIS//DTD DITA Map//EN" "map.dtd">
<map id="birds" xml:lang="en-US">
    <title>Birds</title>
    <topicref href="hummingbird.dita" type="concept"></topicref>
    <topicref href="ostrich.dita" type="concept"></topicref>
    <topicref href="swift.dita" type="concept"></topicref>
</map>
```

This map references three topics.

The first topic is Hummingbird (hummingbird.dita).

```
<?xml version="1.0"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept id="hummingbird" xml:lang="en-US">
    <title>Hummingbird</title>
    <conbody>
        <p>Smallest bird: Dwarf hummingbird (2-1/4 in)</p>
    </conbody>
</concept>
```

The second topic is Ostrich (ostrich.dita).

```
<?xml version="1.0"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept id="ostrich" xml:lang="en-US">
    <title>Ostrich</title>
    <conbody>
        <p>Heaviest bird: Ostrich (330 lb)</p>
    </conbody>
</concept>
```

The third topic is Swift (swift.dita).

```
<?xml version="1.0"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept id="swift" xml:lang="en-US">
    <title>Swift</title>
    <conbody>
        <p>Fastest bird flying: Common Swift (125 mi/hr)</p>
    </conbody>
</concept>
```

2. Prepare the XLIFF file.

Your tool should create a single XLIFF file for all strings in the map and topics. The source strings (in this case) are English. The example shows the completed XLIFF file.

```
<xliff version="1.2">
 <file original="birds.ditamap" source-language="en-US" datatype="xml">
  <body>
   <trans-unit id="birds_t" resname="title">
    <source>Birds</source>
   </trans-unit>
  </body>
 </file>
 <file original="hummingbird.dita" source-language="en-US" datatype="xml">
  <body>
   <trans-unit id="hummingbird_t" resname="title">
    <source>Hummingbird</source>
   </trans-unit>
   <trans-unit id="hummingbird_p" resname="p">
    <source>Smallest bird: Dwarf hummingbird (2-1/4 in)</source>
   </trans-unit>
  </body>
 </file>
 <file original="ostrich.dita" source-language="en-US" datatype="xml">
  <body>
   <trans-unit id="ostrich_t" resname="title">
```

```
   <source>Ostrich</source>
  </trans-unit>
  <trans-unit id="ostrich_p" resname="p">
   <source>Heaviest bird: Ostrich (330 lb)</source>
  </trans-unit>
 </body>
</file>
<file original="swift.dita" source-language="en-US" datatype="xml">
 <body>
  <trans-unit id="swift_t" resname="title">
   <source>Swift</source>
  </trans-unit>
  <trans-unit id="swift_p" resname="p">
   <source>Fastest bird flying: Common Swift (125 mi/hr)</source>
  </trans-unit>
 </body>
</file>
</xliff>
```

3. Once the XLIFF file is ready, send it and a PDF rendering of the source content to your LSP.

   The PDF will help the LSP's translators to understand the context for the translation.

4. Receive the translated XLIFF file from the LSP.

   The following example shows the XLIFF file with both English and German text.

```
<xliff version="1.2">
 <file original="birds.ditamap" source-language="en-US" datatype="xml">
  <body>
   <trans-unit id="birds_t" resname="title">
    <source>Birds</source>
    <target>Vögel</target>
   </trans-unit>
  </body>
 </file>
 <file original="hummingbird.dita" source-language="en-US" datatype="xml">
  <body>
   <trans-unit id="hummingbird_t" resname="title">
    <source>Hummingbird</source>
    <target>Kolibri</target>
   </trans-unit>
   <trans-unit id="hummingbird_p" resname="p">
    <source>Smallest bird: Dwarf hummingbird (2-1/4 in)</source>
    <target>Kleinster Vogel: Zwergkolibri (5,7 cm)</target>
   </trans-unit>
  </body>
 </file>
 <file original="ostrich.dita" source-language="en-US" datatype="xml">
  <body>
   <trans-unit id="ostrich_t" resname="title">
    <source>Ostrich</source>
    <target>Strauß</target>
   </trans-unit>
   <trans-unit id="ostrich_p" resname="p">
    <source>Heaviest bird: Ostrich (330 lb)</source>
    <target>Schwerster Vogel: Strauß (150 kg)</target>
   </trans-unit>
  </body>
 </file>
 <file original="swift.dita" source-language="en-US" datatype="xml">
  <body>
   <trans-unit id="swift_t" resname="title">
    <source>Swift</source>
    <target>Mauersegler</target>
   </trans-unit>
   <trans-unit id="swift_p" resname="p">
    <source>Fastest bird flying: Common Swift (125 mi/hr)</source>
    <target>Schnellster Vogel: fliegend: Mauersegler (200 km/h)</target>
   </trans-unit>
  </body>
 </file>
</xliff>
```

5. After you receive the translated XLIFF file, convert it back to original DITA format.

   After the XLIFF file is transformed back to DITA, the result is an identical German DITA map and topics:

```
<?xml version="1.0"?>
<!DOCTYPE map PUBLIC "-//OASIS//DTD DITA Map//EN" "map.dtd">
<map id="de-birds" xml:lang="de-DE">
  <title>Vögel</title>
  <topicref href="de-hummingbird.dita" type="concept"></topicref>
  <topicref href="de-ostrich.dita" type="concept"></topicref>
  <topicref href="de-swift.dita" type="concept"></topicref>
</map>
```

   The map references these three topics in German.

   The first topic is Kolibri (de-hummingbird.dita).

```
<?xml version="1.0"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept id="de-hummingbird" xml:lang="de-DE">
   <title>Kolibri</title>
   <conbody>
      <p>Kleinster Vogel: Zwergkolibri (5,7 cm)</p>
   </conbody>
</concept>
```

   The second topic is Strauß (de-ostrich.dita).

```
<?xml version="1.0"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept id="de-ostrich" xml:lang="de-DE">
   <title>Strauß</title>
   <conbody>
      <p>Schwerster Vogel: Strauß (150 kg)</p>
   </conbody>
</concept>
```

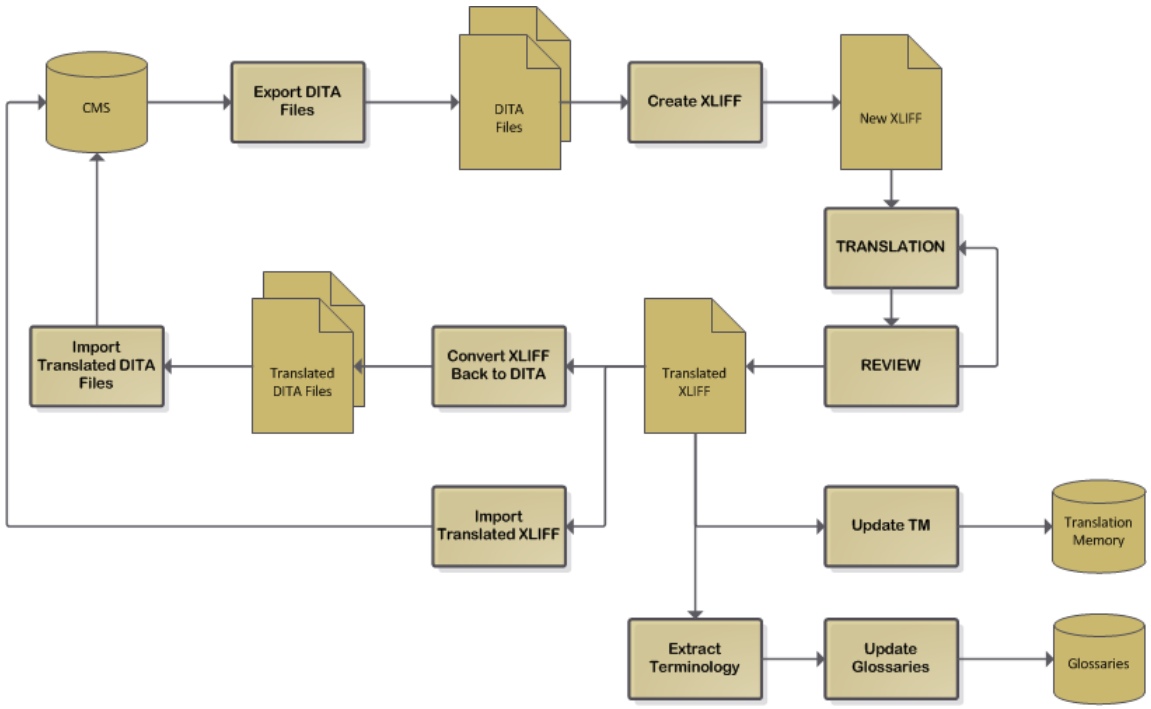
   The third topic is Mauerselger (de-swift.dita).

```
<?xml version="1.0"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept id="de-swift" xml:lang="de-DE">
   <title>Mauersegler</title>
   <conbody>
     <p>Schnellster Vogel fliegend: Mauersegler (200 km/h)</p>
   </conbody>
</concept>
```

   You should have a directory structure for the translated content that is identical to the original directory of the source content.

6. If your images are not in SVG format, you must copy them to the new translation project structure before you can render the translated project.
7. Export the translations in your XLIFF file to TMX (Translation Memory eXchange) format and update your Translation Memory.
8. Store the translated XLIFF and the TMX file in your content repository. Both files will play a crucial role in the maintenance workflow.

The following workflow diagram illustrates the initial process, including some optional steps not described above:



Maintenance workflow

As products and processes are updated, you will update some of your topics, write new ones, and need to update your translations. At that point, you will see the benefits of translation reuse with these techniques:

- reuse In-Context Exact (ICE) matches
- recover translations of similar text from Translation Memory (TM)
- generate updated translations using Example-Based Machine Translation (EBMT)

The maintenance workflow proceeds as follows:

1. Convert the updated DITA map and topics to XLIFF.
2. Using your translation tool, compare the new XLIFF with the one you previously had translated and recover In-Context Exact (ICE) matches.

   This step recovers the translations of text that has not changed since last translation cycle.
3. After recovering all ICE matches, mark all translated segments as untranslatable ("do not translate") in the translation tools.

   Translations will remain visible in the XLIFF as context information for the translator but will not need to be changed.
4. If you have not yet updated your Translation Memory with the translations from the previous cycle, import the TMX into the Translation Memory of your translation environment.
5. Use your TM engine to retrieve matches for the segments that remain untranslated.

   A TM engine can evaluate the similarities between current text requiring translation and entries that exist in its database. A match is a *perfect* match when source text is exactly the same as the text found in the translation memory. Entries identical to the text being translated are considered *perfect* matches; a match is *fuzzy* when source text is similar but not 100% equal to the text found in the database.

6. If your translation memory only contains entries from a very similar project, you may want to accept all *perfect* matches as final.

   Because there is no guarantee that these matches are the right translations, you should let professional translators approve them. Nevertheless, you may ask your LSP to set a special price for segments with good matches from your own TM.

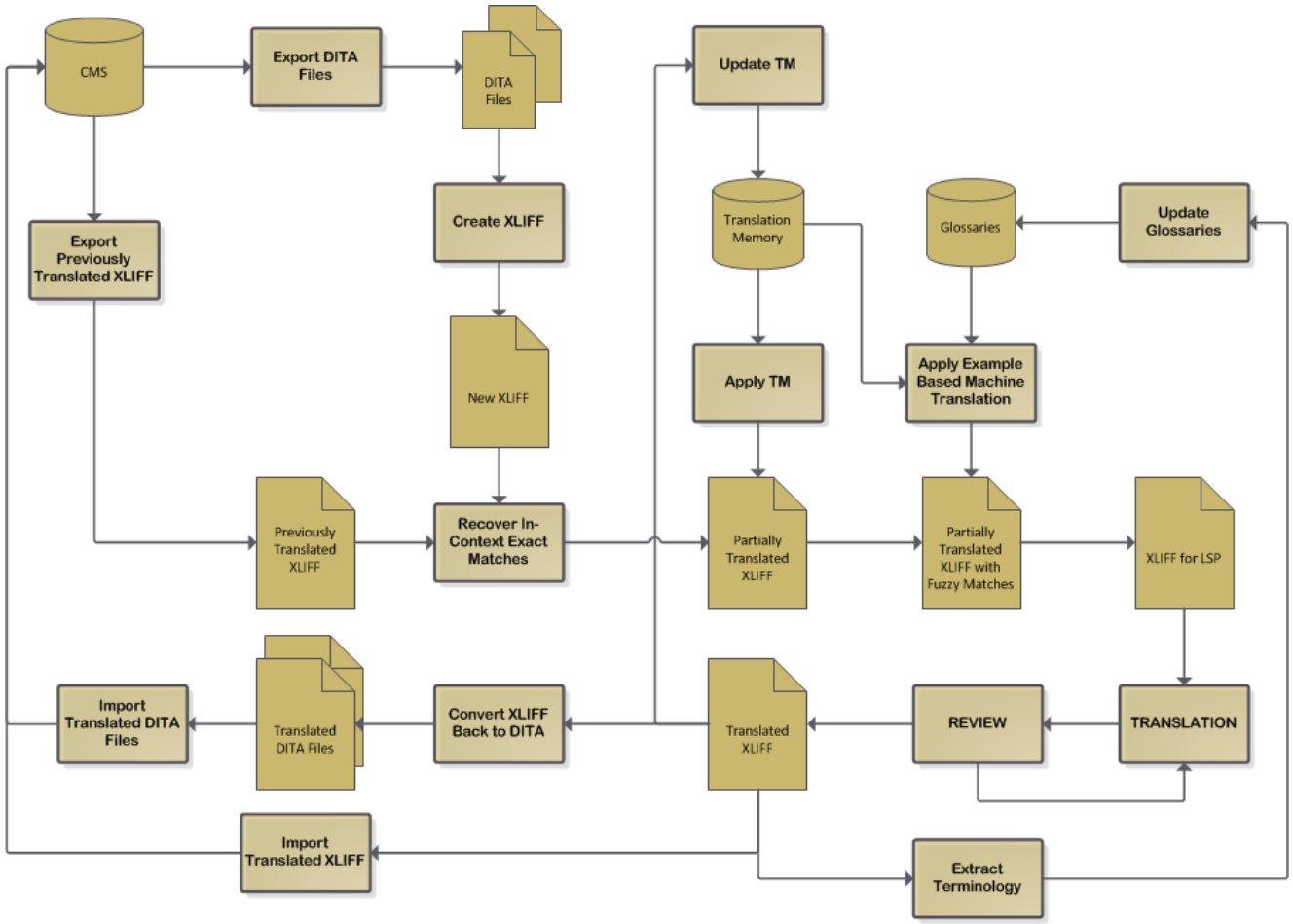
7. Use EBMT and recover additional matches.

   Sometimes the difference between the old text and the new one is simply an updated number. Such a small change is something a good CAT (Computer-Aided Translation) program can correct automatically using EBMT techniques. An EBMT engine can also automatically correct the translation of known terms with the aid of a terminology database.

   You should now have an XLIFF file ready to send to an LSP for completing the translation cycle.

8. Send the XLIFF file, partially translated via the preceding steps, with an updated PDF rendering.
9. Receive back the translated XLIFF file and convert it back to the DITA map and topics.
10. Remember to update your TM engine when you receive the translated version back.
11. Finally, if your translation budget allows it, generate a PDF rendering of the translated project and send it with a copy of the translated XLIFF to your LSP for proofreading.
    If the reviewer finds an error, it can be corrected in the XLIFF file and sent back to you to update your topics and your TM.

The full maintenance workflow is shown in the following diagram:



Resources

Download or view the latest approved *DITA Specification*.

Read the *XLIFF 1.2 Specification*, which defines the XML Localization Interchange File Format (XLIFF). The purpose of this vocabulary is to store localizable data and carry it from one step of the localization process to the next, while allowing interoperability between tools.