xliffRoundTrip Tool user guide

License for xliffRoundTrip Tool

Publication date: May 2009

Bryan Schnabel

CONTENTS

Topics Covered in this guide

This document covers three topics:

- Overview of xliffRoundTrip Tool
- Overall dependencies
- How to use the xliffRoundTrip graphical user interface
- How to use the xliffRoundTrip command-line tool (command-lineXRT.0.7.jar)
- How to compile jar files from source

CHAPTER

1

Overview of xliffRoundTrip Tool

The xliffRoundTrip tool in an open source tool (http://sourceforge.net/projects/xliffroundtrip/) that makes use of the XLIFF (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff) open standard. It enables a user to perform a round trip between any well-formed XML, to XLIFF, and back to the well-formed XML.

There are three ways to use the xliffRoundTrip Tool:

- You can use the Graphical User Interface by accessing the xliffRoundTrip.0.7.jar

  More details on the GUI version are listed here.

  This interface lets you browse to the XML file you wish to convert to XLIFF, or to the converted XLIFF file you wish to convert back to XML.

  Note

  Only XLIFF files converted via xliffRoundTrip Tool are likely to convert back to XML smoothly.

- You can use the Command Line interface (command-lineXRT.0.7.jar).

  More details on the Command Line version are listed here.

- You can compile your own executable JAR files via the included Java source code.

  More details on the Source Code are listed here.

Note

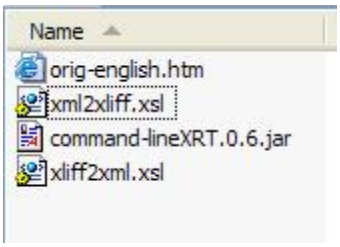
Before testing the tool, you might want to check out the Known Bugs at the end of this document.

Here's an actual round trip (the command-line way):

1. Start with this well formed XML, orig-english.htm:

```
<html>
 <h2><font color="green">This is the<b> YUMMY TREATS</b> web
    site</font></h2>
 <p>You can learn everything about<b> YUMMY TREATS</b>
   here.</p>
 <p>Buy a box today!</p>
</html>
```

2. Make sure the input XML file, the command-lineXRT.0.7.jar, and the two XSLT files, xml2xliff.xsl, and xliff2xml.xsl are in the same directory. Like this:



3. Fire off this command:

```
java -jar command-lineXRT.0.7.jar orig-english.htm xml2xliff.xsl > english-english.xlf
```

This will output an XLIFF file called english-english.xlf. It will look like this:

```
<xliff xmlns="urn:oasis:names:tc:xliff:document:1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xmrk="urn:xmarker" ver
 <file original="html" source-language="en" datatype="plaintext">
  <header>
   <xmrk:nest>
    <xmrk:html xmarker_idref="html-0">
     <h2 xmlns="urn:xmarker" xmarker_idref="h2-1">
      <font color="green" xmarker_idref="font-2">
       <b xmarker_idref="b-3"/>
      </font>
     </h2>
     <p xmlns="urn:xmarker" xmarker_idref="p-4">
      <b xmarker_idref="b-5"/>
     </p>
     <p xmlns="urn:xmarker" xmarker_idref="p-6"/>
    </xmrk:html>
   </xmrk:nest>
  </header>
  <body>
   <group id="N10001axmarkhtml-0">
    <group id="N10003bxmarkh2-1">
     <trans-unit id="font-2">
      <source>This is the<g id="b-3"> YUMMY TREATS</g> web
     site</source>
      <target>This is the<g id="b-3"> YUMMY TREATS</g> web
     site</target>
     </trans-unit>
    </group>
    <trans-unit id="p-4">
     <source>You can learn everything about<g id="b-5"> YUMMY TREATS</g>
           here.</source>
     <target>You can learn everything about<g id="b-5"> YUMMY TREATS</g>
           here.</target>
    </trans-unit>
    <trans-unit id="p-6">
     <source>Buy a box today!</source>
     <target>Buy a box today!</target>
    </trans-unit>
   </group>
  </body>
 </file>
```

```
</xliff>
```

Notice the xliff file contains <source and <target tags that each contain English text.

4. Translate the target strings to Reverseze, like this, and call it English-Reverseze.xlf:

```
<xliff xmlns="urn:oasis:names:tc:xliff:document:1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xmrk="urn:xm
 <file original="/x-" source-language="en" datatype="plaintext">
  <header>
   <xmrk:nest>
   <xmrk:html xmarker_idref="html-0">
    <h2 xmlns="urn:xmarker" xmarker_idref="h2-1">
     <font color="green" xmarker_idref="font-2">
      <b xmarker_idref="b-3"/>
     </font>
    </h2>
    <p xmlns="urn:xmarker" xmarker_idref="p-4">
     <b xmarker_idref="b-5"/>
    </p>
    <p xmlns="urn:xmarker" xmarker_idref="p-6"/>
   </xmrk:html>
   </xmrk:nest>
  </header>
  <body>
   <group id="N10001axmarkhtml-0">
   <group id="N10003bxmarkh2-1">
    <trans-unit id="font-2">
     <source>This is the<g id="b-3"> YUMMY TREATS</g> web
 site</source>
     <target>Tsrh rh gsv<g id="b-3"> YUMMY TREATS</g> dvx
 hrgv</target>
    </trans-unit>
   </group>
   <trans-unit id="p-4">
    <source>You can learn everything about<g id="b-5"> YUMMY TREATS</g>
    here.</source>
    <target>Ylf yzm ovzim vevibgsrmt zxlfg<g id="b-5"> YUMMY TREATS</g>
    sviv.</target>
   </trans-unit>
   <trans-unit id="p-6">
    <source>Buy a box today!</source>
    <target>Bfb z xlc glwzb!</target>
   </trans-unit>
   </group>
  </body>
 </file>
</xliff>
```

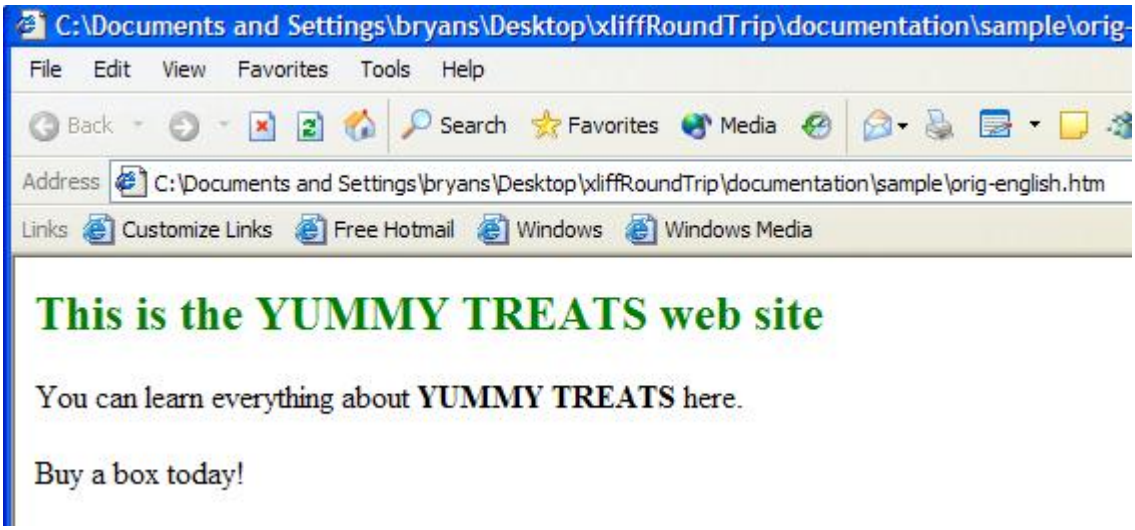
5. Fire off this command.

```
java -jar command-lineXRT.0.7.jar English-Reverseze.xlf xliff2xml.xsl > new-Reverseze.htm
```

You will now have a new file with the exact same structure as the original; but now it's Reverseze, like this:

```
<html xmlns:xmrk="urn:xmarker" xmlns:xlf="urn:oasis:names:tc:xliff:document:1.2">
 <h2>
  <font color="green">Tsrh rh gsv<b> YUMMY TREATS</b> dvx
  hrgv</font>
 </h2>
 <p>Ylf yzm ovzim vevibgsrmt zxlfg<b> YUMMY TREATS</b>
    sviv.</p>
 <p>Bfb z xlc glwzb!</p>
</html>
```

Before


C:\Documents and Settings\bryans\Desktop\xliffRoundTrip\documentation\sample\orig-
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\Documents and Settings\bryans\Desktop\xliffRoundTrip\documentation\sample\orig-english.htm
Links Customize Links Free Hotmail Windows Windows Media
This is the YUMMY TREATS web site
You can learn everything about YUMMY TREATS here.
Buy a box today!

After


Mozilla Firefox
File Edit View History Bookmarks Tools
file:///C:/presentations/schnabel_demos_unde
Most Visited Getting Started Latest Headlines Customer Documentat...
Tsrh rh gsv YUMMY TREATS dvx hrgv
Ylf yzm ovzim vevibgsrmt zxlfg YUMMY TREATS sviv.
Bfb z xlc glwzb!

CHAPTER

2

Overall components and dependencies

This program has three components:

Compile Program Files

- command-lineXRT.0.7.jar
- xliffRoundTrip.0.7.jar

Optional Source Code

- xrtV07.java
- MANIFEST1.MF
- MANIFEST2.MF

Optional XML Schemas for XLIFF validation

- xliff-core-1.2-strict.xsd
- xmarker.xsd

2.1 Dependencies

Sun’s JAXP 1.1 API (or later), http://java.sun.com/xml/jaxp/downloads/index.html

2.2 CLASSPATH requires

- crimson.jar
- jaxp-api.jar
- xalan.jar

CHAPTER

3

How to use the xliffRoundTrip graphical user interface

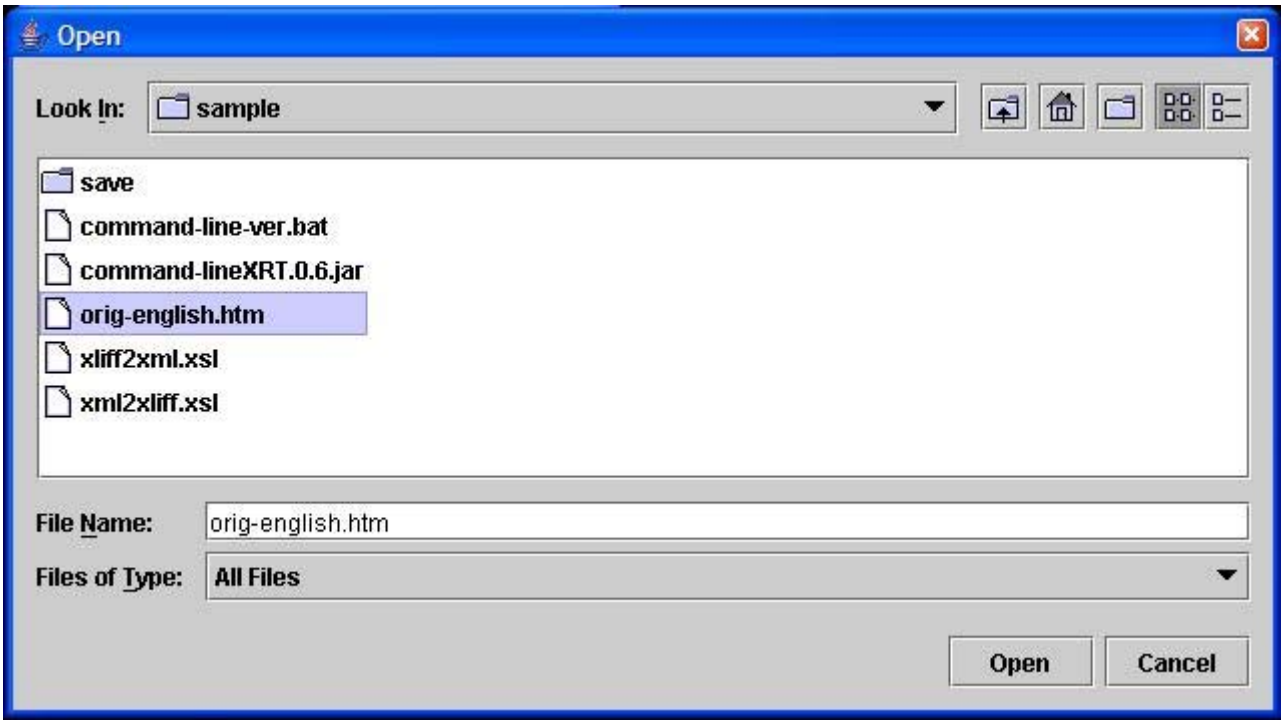
Here's how to use the graphical user interface. Please bear in mind that my primary focus is ensuring the XSLTs are working. This Java interface is kind of fun to develop, but it receives far less attention and development time. But here goes.

1. Ensure that the XSLT files and the xliffRoundTrip.0.7.jar file are in the same directory.
2. Fire off this command:

   java -jar xliffRoundTrip.0.7.jar

   It will bring up the user interface. It looks like this:

3. Click the "Choose file XML" button (bug note: click to the "all file types" choice, to see all files).
4. Navigate to the file you want to transform to xliff.



5. Click the “Open” button.

   You will get an english-english xliff file. The file will have the same name, with a .xlf extension added. In this case, orig-english.htm.xlf.

6. Translate the English <target> strings to some language

7. Perform the same operation, but pick the “Choose XLIFF” button.

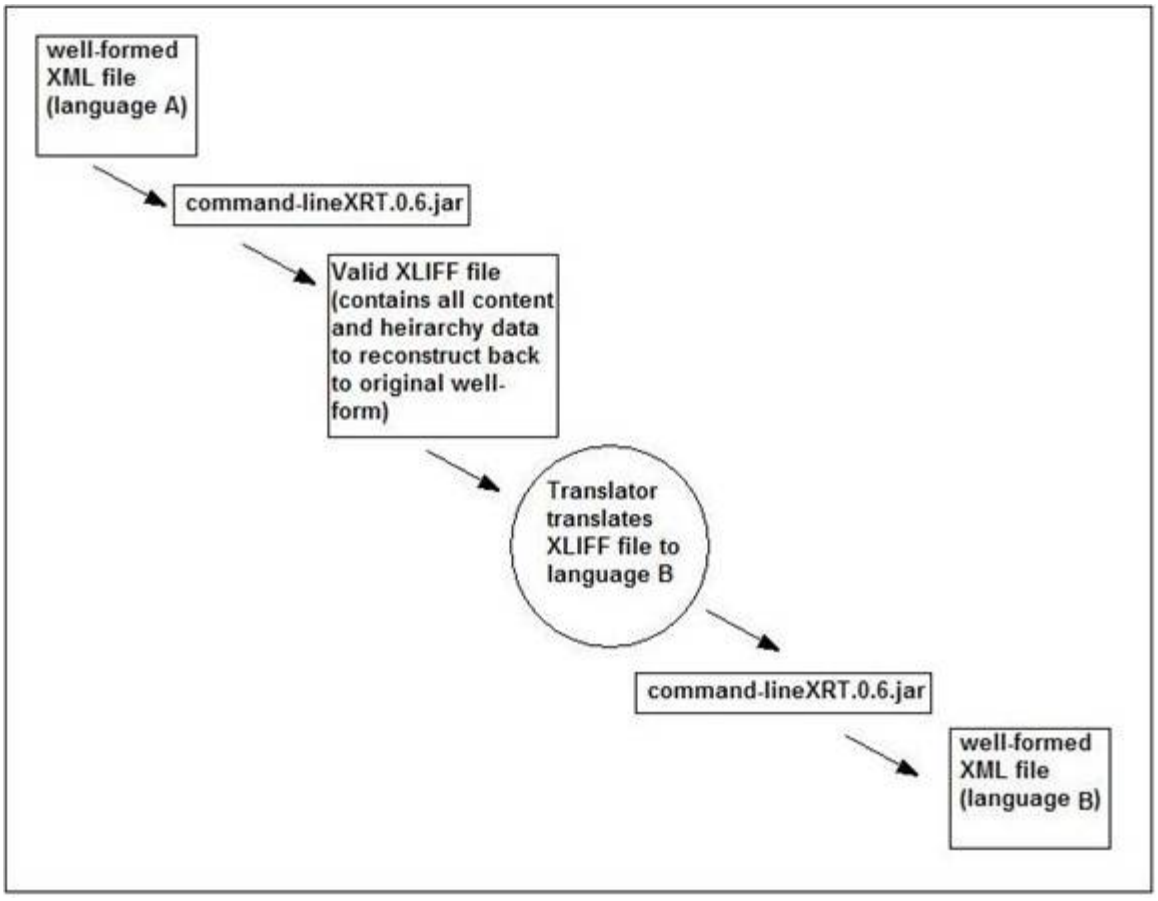
   It will transform the file back to its original format, but it will be French. The name will be the same, but with a .xml extension. In this case, orig-english.htm.xlf.xml.

CHAPTER

4

How to use the xliffRoundTrip command-line tool (command-lineXRT.0.7.jar)

Here's how it works:



For now, be sure the following files are in the same directory:

- command-lineXRT.0.7.jar
- xliff2xml.xsl
- xml2xliff.xsl
- [source.xml] - this is your well-formed xml file, compliant with the temporary restrictions listed below.

Fire off this command: java -jar command-lineXRT.0.7.jar [source.xml] xml2xliff.xsl > [target.xlf]

Note

[source.xml] and [target.xlf] are names you supply.

This will result in an xliff file, that you named [target.xlf].

Next, have a translator translate the XLIFF trans-unit target strings. Keep the XLIFF file valid and well-formed.

Then comes the transformation back to its original doctype: Fire off this command:

java -jar command-lineXRT.0.7.jar [translated.xlf] xliff2xml.xsl > [translated.xlf]

This will result in a translated xml file.

Note

If you have a favorite XSLT processor you want to use instead of the xliffRoundTrip.java, you can just use it with the xsl files included in this distribution.

CHAPTER

5

How to compile jar files from source

Here’s how to compile the java files, in order to create the jar files.

Ensure that the following files are in the same directory: xrtV07.java, MANIFEST1.MF, and MANIFEST2.MF.

Fire off the following command:

javac xrtV07.java

This will create the class files.

5.1 How to compile the GUI version

Fire off the following command:

jar cmfv MANIFEST1.MF xliffRoundTrip.0.7.jar *.class

5.2 How to compile the command line version

Fire off the following command:

jar cmfv MANIFEST2.MF command-lineXRT.0.7.jar clxrtV07.class

CHAPTER

6

Known Bugs

The following bugs are being worked on (anyone who wants to advise me on fixes will receive heartfelt gratitude):

- XML file cannot contain a doctype declaration with a PUBLIC identifier. I've found that just commenting it out works fine. The commented doctype declaration makes it through the whole roundtrip, and can be easily uncommented after the trip.
- XML file cannot contain namespaces. This should not be hard to fix. I just need some time.
- In the file chooser step (with the GUI version), it works better to set the file type to "all files."

Please let me know if you find any other bugs (I'm sure they're in there)