Using DITA for Successful Localization

By its nature, content written for reuse is more structured, and is therefore approached with a closer eye for a uniform writing style and branding.

Using the DITA standard can bring enormous improvements in the efficiency of localizing documentation. Some organizations report efficiency gains of 30-50% over traditional desktop publishing systems, and use the savings to expand further into global markets. While these numbers are compelling, localized DITA implementation is still a mystery to many potential adopters. The big picture of how it all works is not obvious, and the details affecting the quality and/or cost of localization are numerous.

This article addresses the questions of "how does it work?", "what do I need to plan for?" and "what are the gotchas?" Although DITA implementation strategies vary and it is always necessary to adapt and test a localization system, this article can provide a starting point for adaptation and a framework for testing. Some of the concepts in this article also apply to documents written in XML languages other than DITA.

This article covers moderately advanced concepts, and assumes that you already have an understanding of concepts such as *DITA map, DITA topic, attribute, element, style sheet, conditional text, conref, specialization, and separation of content and formatting.*

How Localization Works in DITA Projects

One of the most frequently-asked questions about localization and DITA is simply, "how do you do it?" In particular, when discussing DITA features that enable content to be re-used in different documents, people ask how these features interact with translation. To understand the big picture, consider a scenario in which you write a document in English, and then have it translated into Japanese. A high-level overview of the process appears in Table 1.

Table 1: Example of steps for producing localized DITA documents

Step	Description	Notes
1	You create your content in English, using a DITA authoring tool	Most teams choose to have one file per topic, organized into folders. You will probably also have image files, map files indicating the topic hierarchy, and a primary map file indicating the hierarchy of maps.
2	When English content is ready for translation, you send a copy of the relevant files to the translator.	If you use a translation memory system, you could either send your content to translators through that system, or zip them up and email them. If you do not use a translation memory system, you will probably zip up and email your files. You can send a subset of your files instead of sending all files at once. However, keep files in their folder structures so that the links between them don't break; tools are available to automate this process. If you are using a content management system (CMS), ask your CMS vendor what to send to the translators.
3	The translator opens the English files in their system and replaces the English content, sentence by sentence, with Japanese content.	If you are working with a translation memory system and the content has been translated before, the translator will work only with the new content at this stage. The translator does not change any DITA markup, except to set an attribute indicating that the content is now in Japanese, and possibly to translate some attribute values such as "navtitle". The translator does not change any file or folder names. As markup and file names are left intact, any use of conditional text, conrefs, cross-references, and other types of links between documents will be automatically implemented in the Japanese content.

4	The translator sends the Japanese files back to you.	The translator should send you files in the same folder structure that you used in step 2.
5	You put your English content into a DITA publishing system. The system produces English deliverables.	DITA publishing systems typically require customization in order to produce deliverables with your organization's look-and-feel. However, once a system is set up, it should produce a professional-looking publication automatically, and not require you to tweak the output at this time.
6	You put your Japanese content into a DITA publishing system. The system produces Japanese deliverables.	Typically, the publishing system for Japanese will be the same as the one used for English, but it will do certain things differently if the content includes an attribute saying that it is in Japanese. For example, a publishing system can be configured to use Japanese fonts for Japanese content.

For many organizations, the ability to generate production-quality output for each language in one button-click is the chief reward of using DITA, as the traditional desktop publishing paradigm requires labor-intensive formatting whenever content is translated.

A DITA publishing system, which is often based on the open-source *DITA Open Toolkit*, converts DITA content into human-readable formats such as online help systems, HTML for websites, and PDF for books and brochures. The functions of a DITA publishing system include assembling content from multiple files, inserting page breaks at appropriate points, and building a table of contents. Publishing features relevant to localization may include the following:

- Generated text: The system ensures that generated text, such as "Note:" at the beginning of <note> elements for English and "Nota:" at the beginning of <note> elements for Spanish, is appropriate for each language.
- Index and glossary sorting: Indexes and glossaries are sorted according to language-specific rules.
- Help system UI: If the output is an online help system, text on the help system's buttons and tabs appears in the correct language.
- Fonts: The publishing system should be configurable to use an appropriate font for each language, as no single font is optimal for all languages.
- Language-specific styling conventions: For example, in English software manuals it is common to display names of menu items in bold. In Japanese, the convention is to put brackets around menu item names.
- Text direction: Some languages, such as Arabic and Hebrew, must be displayed right-to-left instead of left-to-right.

Identifying the Language for Your Content

To take full advantage of DITA publishing systems, you must indicate the language of the content to the publishing system.

Some aspects of output vary by region as well as by language. Among the English-speaking regions, there are for practical purposes no differences in how output should be generated, even though there are often differences in the content itself. However, for some languages it does matter – for example, index sorting rules can vary by region.

Use the "xml:lang" attribute to indicate the language of the content. For example, the following <map> element has a xml:lang attribute, indicating that publishing systems should display it in a way that is optimal for the French language as used in France:

For many organizations, the ability to generate production-quality output for each language in one button-click is the chief reward of using DITA, as the traditional desktop publishing paradigm requires labor-intensive formatting whenever content is translated.

<map xml:lang = "fr-fr"><title>Guide de l'utilisateur</title>…</map>

Sometimes there is more than one possible language code for what is essentially one language. For example, some people designate Simplified Chinese with the language code "zh-cn", whereas others use the language code "zh-Hans" for the same thing. The language codes used in the xml:lang attribute must match the language codes used in your publishing system. The DITA OT is preconfigured to work with the language codes listed here: http://dita-ot.sourceforge.net/doc/ot-userguide/xhtml/localizing/aboutlocalizing.html . For example, you can see from that page that the DITA OT expects Simplified Chinese content to be tagged with "zh-cn".

If you don't set the xml:lang attribute anywhere and do not use a publishing parameter to indicate the language, publishing systems should assume the content is in English. In any document that is in a language other than English, set the xml:lang attribute at the root of each map file and each topic file.

Creating Localization-Friendly Content

A Google search for "writing for localization" yields many good guidelines on writing in a style that will make localization better and less expensive. *Controlled language software* (http://en.wikipedia.org/wiki/Controlled_natural_language) is available for use with DITA editors, and is an increasingly popular way to enforce use of a smaller vocabulary and clearer writing style.

The rich capabilities of DITA mean that there are some additional principles to keep in mind, beyond how sentences are crafted. The following sections describe some specific issues with various types of content.

Setting up Map and Topic Files

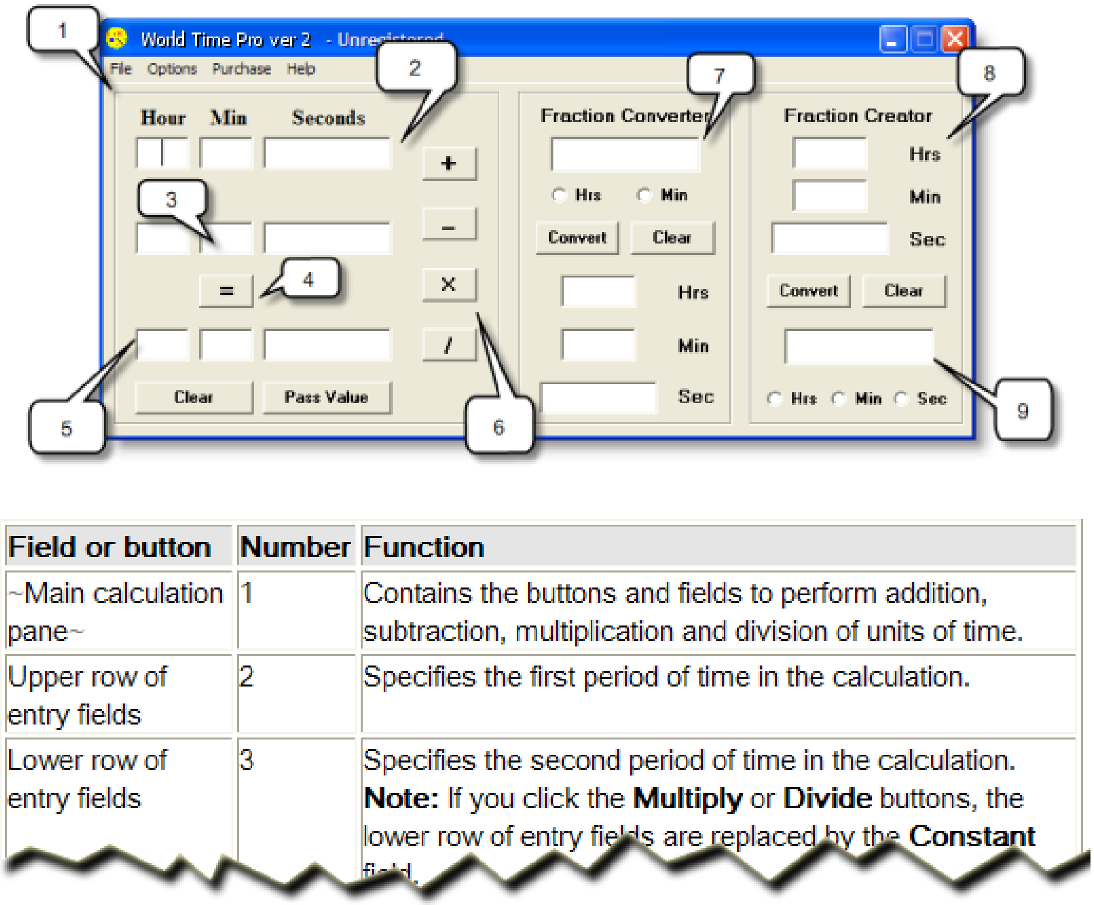
In DITA map files, you can put the title of the document either in a <title> element within the <map> element, or in a "title" attribute on the <map> element. In general, a good practice is to put translatable content only in elements, as some translation systems do not allow translation of attribute values. Furthermore, as attributes are critical for linking and conditional text in DITA, it is a good idea to discourage translators from translating attributes.

For headings, it is possible to put the heading text in the "navtitle" attribute of a <topichead> element. As an alternative that does not require putting translatable content into attributes, create a <topicref> pointing to a topic, put the heading text in the topic title, and leave the topic body blank.

Images and Callouts

A common requirement is to re-use images across multiple languages, but translate the callouts or other annotations that you want the reader to associate with the image. One way to meet this requirement is to put numbers rather than callouts in the image file, and add a legend below the image. Translators then translate the legend as they would any other part of the topic.

Figure 1: Numbered callouts with a legend



Field or button	Number	Function
~Main calculation pane~	1	Contains the buttons and fields to perform addition, subtraction, multiplication and division of units of time.
Upper row of entry fields	2	Specifies the first period of time in the calculation.
Lower row of entry fields	3	Specifies the second period of time in the calculation. **Note:** If you click the **Multiply** or **Divide** buttons, the lower row of entry fields are replaced by the **Constant** field.

A more sophisticated approach is to use the SVG format for images. SVG can encapsulate a raster image along with editable callout text. You can edit SVG files in applications such as Adobe® Illustrator, and send the SVG files to your translator to have the callouts translated. If you want to use SVG for images, make sure that your authoring tool and your publishing system can display SVG images in all of the output formats you require.

If you are using SVG, remember to not crowd English text into the image. Leave extra room, as some languages require 30% more space to convey the same information.

Image “Alt Text”

DITA allows you to put alternative text for <image> elements in either an element or an attribute. Again, it’s best to put translatable content only in elements, so use the <alt> element. When a DITA publishing system generates HTML output, it should put the contents of the DITA <alt> element into the HTML “alt” attribute.

Index Markers

For most languages, translators can simply translate the contents of <indexterm> elements, and the DITA publishing system should sort the index correctly. However, some additional work is needed for Japanese and Chinese indexes.

For Japanese translations, ask translators to insert an <index-sort-as> element in each <indexterm> element. It is needed because Japanese terms are alphabetized differently depending on the context in which the term is used.

For Chinese translations, publishing systems sometimes require an <index-sort-as> element in every <indexterm> element. However, translators can leave the <index-sort-as> elements empty, as they can be filled in automatically by the publishing system.

Before sending content to be translated into Japanese or Chinese, you might want to run a script that automatically inserts an empty <index-sort-as> element into every <indexterm> element.

Glossary Entries

Some DITA publishing systems can automatically sort glossary entries, which is a valuable feature. For most languages, sorting is automatic, however if one of your required languages is Japanese, some setup work is needed.

At this time there is no way to automate sorting of glossary entries in Japanese if you are using only the base DITA element types, because there is no element type for glossaries that is equivalent to the <index-sort-as> element. To sort Japanese glossary entries, you must create a specialized <glossary-sort-as> element and place one within each <glossentry> element, then build a feature into your publishing system to sort based on the specialized <glossary-sort-as> element. A proposal exists for future versions of the DITA standard to include a <glossary-sort-as> element so that specialization will be unnecessary.

Inline Elements

To ensure clean alignment of translated content, avoid extra spaces at the beginning or end of inline elements such as <indexterm>, <uicontrol>, and <menucascade>.

In general, using semantic element types, rather than element types which directly specify formatting, can give you more options for optimal localization. For example, as conventions vary on how to format the names of software buttons and menu items, use the <uicontrol> element rather than the <b> element for these names.

Using Conditional Text and Content References

The *conditional text* feature in DITA makes it possible to produce documents customized for different audiences, or different product variations, from one set of source files. A suggested guideline is to conditionalize only whole sentences. Conditionalizing fragments within a sentence is problematic because in some languages the rest of the sentence might need to change depending on which condition is in effect.

The *conref* (content referencing) feature in DITA makes it possible to use a fragment of content in a variety of contexts. As with conditional text, it is risky to use a conref for a fragment smaller than a sentence, with the exception of proper nouns such as product names.

Identifying Content to Not Translate

With DITA you can effectively communicate to translators when you have parts of a document that should not be translated. There are two ways to do this: through the “translate” attribute, and through element types. You can set a “translate” attribute on most DITA elements to indicate that the element should not be translated. For example, if a table cell contains a mailing address that should always remain in its original language, set the “translate” attribute on the table cell to “no”. You can also give translators a list of element types whose contents should not be translated. For example, <codeblock> and <codeph> elements should typically not be translated.

Working with Translators

Translation companies are increasingly experienced in working with XML content, or are at least willing to try. This is fortunate, as rule number one is: *Do not even think of hiring a translator who will not work with DITA XML*. You cannot gain the efficiencies of XML-based publishing if you have to constantly convert your source content between DITA XML and another format such as XHTML or Microsoft® Word files.

Here are some guidelines for working smoothly with your translator:

- Tell the translator what version of DITA you are using, e.g. 1.0 or 1.1.
- If you are using a DITA specialization, send the translator your DTDs or XSDs and your catalog files. Also send a description of the purpose of any element types you have created through specialization.
- Before you send files out for translation, fix all validation errors and broken links, and remove extra spaces.
- Set the expectation that the files the translator sends back to you must validate against your DTDs or XSDs, must not have broken links, and must preserve structural markup. For example, if you send out content containing <uicontrol> elements, the translator should not send you back content with <b> elements in the place of <uicontrol>.
- Decide if you or the translator will set the xml:lang attribute on the DITA map and/or topics after they have been translated. You can set up a script to set this attribute.
- Give translators a list of element types that should never be translated, and make sure they also know about the “translate” attribute.
- Tell translators to never translate content that is in attributes, except the navtitle attribute in <topicref> or <topichead> elements if you use it. If you do use the navititle attribute, make sure your translator *will* translate it.

As an alternative to sending hundreds of topic files to translators, you can package up a set of DITA files into a single file in the XML Localization Interchange File Format (XLIFF) standard. Some translation vendors prefer this format to individual files. Further information on DITA and XLIFF is here: http://wiki.oasis-open.org/xliff/XLIFF-DITA .

Choosing and Setting up DITA Tools

The tools in a typical DITA implementation include a DITA authoring tool for writing, a DITA publishing system for producing output, often a CMS and a translation memory system, and sometimes other technologies. To avoid problems with multilingual character sets, make sure that all parts of your toolset are Unicode-compliant. Make sure your authoring tool enables you to produce localization-friendly content according to the above guidelines, such as being able to set the “translate” attribute on specific elements.

Some typical localization features of DITA publishing systems were described earlier in this article. For publishing systems based on the DITA OT, a description of how to configure these features is available at http://forums.xmetal.com/index.php/topic,951.0.html. If you are considering a publishing system other than the DITA OT, evaluate whether it can accommodate the listed features for your required languages.

Start testing your publishing system early in your project, and test *each required language in each output format*. If your publishing system does not include generated text strings for some of your required languages, locate the appropriate string files in English and send them for translation. A translator or language expert can also help you select appropriate fonts and specify how indexes should be sorted.

Note that at this time the DITA OT lacks appropriate default settings for PDF output in most non-English languages, and configuration is often necessary in order to produce PDF output that can be read at all. HTML output from the DITA OT is better by default for most languages.

Many organizations report that setting up a publishing system is the most challenging aspect of implementing DITA. Engaging a consultant who is experienced in producing localized output from DITA is often very worthwhile, and is a small investment considering the potential benefits of implementing a DITA-based localization system.

Further Reading

The OASIS DITA Translation Subcommittee has produced several documents on optimizing DITA content for translation, available at http://dita.xml.org/wiki/optimizing-dita-for-translations .

Selected Case Studies

- Increasing Revenue while Reducing Costs with Translation Management Technology by Information Builders: http://www.sdl.com/en/Images/GIM%20Presentation_tcm16-27202.swf .
- From Planning to Publishing: How Business Objects Migrated Documentation to DITA One Step at a Time by Business Objects (now SAP): http://www.slideshare.net/abelsp/from-planning-to-publishing-how-business-objects-migrated-documentation-to-dita-one-step-at-a-time .

Acknowledgements

Content shown in Figure 1 was created by Heather Wymer, and is licensed under a Common Public License: http://www.opensource.org/licenses/cpl1.0.php. The author wishes to thank France Baril, Yas Etessam, Wanda Phillips, Tom Magliery, and Murray Smith for their assistance with this article.

Su-Laine Yeo,
JustSystems

JUSTSYSTEMS.

http://www.justsystems.com

North America:
Phone: +1.866.793.1542
Email: contactsales-na@justsystems.com

Europe:
Phone: +44.0.1462.89.082
Email: contactemea@justsystems.com

International: +1.604.697.8705

About JustSystems

JustSystems is a leading global software provider with three decades of successful innovation in office productivity, information management, and consumer and enterprise software. With over 2,500 customers worldwide, the company is continuing a global expansion strategy based on its XMetaL content lifecycle solutions. JustSystems is one of the 2008 KMWorld 100 Companies that Matter in Knowledge Management, a 2008 EContent 100 member, and was recognized on the 2008 KMWorld Trend-Setting Product list for XMetaL. Major strategic partnerships include IBM, Siemens and EMC.