An OASIS White Paper

Best Practice for Indexing DITA Topics for Translation

By JoAnn T Hackos
For the OASIS DITA Translation Subcommittee

4 March 2008

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 purpose of the OASIS DITA Technical Committee (TC) is to define and maintain the Darwin Information Typing Architecture (DITA) and to promote the use of the architecture for creating standard information types and domain-specific markup vocabularies. The Translation Subcommittee defines best practices and guidelines for DITA authoring, translation and localization, and recommends solutions for industry requirements for consideration by the OASIS DITA TC. The group recommends widespread adoption of these concepts through liaisons with industry, other standards, and providers of commercial and open source tools.

Table of Contents

1. Statement of the Problem

In the DITA model, Index <indexterm> tags may be inserted inline (meaning before, after, or within sentences or phrases) in most DITA elements, including the block elements of paragraph, bulleted list, and numbered list. They are used to create in-situ index references to the topic or to text within the topic.

However, inserting an inline element such as <indexterm> anywhere into a block element may present problems for translation. In addition, translation memory (TM), which stores segments that have been previously translated, may not match segments that are divided by index tags, especially if the index tag has been added or moved since the previous translation.

In the past, it has been common practice for writers using desktop publishing tools to insert index tags anywhere in a sentence. Indexing tools, such as those available in traditional desktop publishing applications, allow the writer to select a term or phrase in the text and use it to create an index item without retyping the term in the index tag. In this case, the text remains in place and also generates an index item. Indexing in DITA XML must be handled differently. After a word or phrase has been enclosed in the <indexterm> tag, it is no longer recognized as part of the text of the block element.

Note that the recommendations in this document apply specifically to indexing in DITA. However, many of the comments herein apply to indexing any XML-based text.

2. Recommended Best Practices

If you keep the following points in mind, you should be able to ensure that your insertion of index tags has the least impact on the translation process and does not increase the cost of your translations:

- Insert index entries that refer to entire topics in the prolog element using the <keywords> tag (<prolog><metadata><keywords><indexterm> word(s) to be included in the index</indexterm></keywords></metadata></prolog>). Index entries using the <keywords> tag in the prolog are processed to refer to the page that contains the title of the topic. Index entries referencing the title of a topic may also be placed on the topic reference in the DITA map (<topiref><topicmeta><keywords><indexterm>).[Example 1]
- Insert block-level index tags immediately following the start tag of the applicable containing block element. [Examples 2 and 2a]
- Insert a sentence-level index tag immediately preceding the sentence in which the text occurs. [Example 3]
- Insert word or phrase-level index tag immediately preceding the text to which it applies. You must retype the word or phrase in the <indexterm> element. If you tag an existing word or phrase as an <indexterm>, it will not appear in the final text of the block. [Example 4]
- Avoid relocating an index tag within a text that has not changed since the last translation when you have pointed to a specific word or phrase in the text. A relocated index term generates a fuzzy match because the segmentation of the sentence changes from that stored in the translation memory.
- If your index will be translated into languages that require per-word modifications to the sort order, such as Japanese, insert the <index-sort-as> tag after each index entry so that the translator can specify the sort order required in the target languages. [Example 5]

When the <indexterm> element is encountered within a block element, the translation tool should generate a subflow. The subflow text is separated from the text of the block element and analyzed separately. The element is replaced by a marker in the text, and its processing is delayed.

Example Markup for Index Entries

The examples below illustrate the recommended placement of index tags in DITA topics.

Example 1--Insert index entries that refer to entire topics in the prolog element

The following example shows an index term inserted in the prolog.

<concept>

<title>Introduction to topic-based authoring</title>

<shortdesc>The DITA architecture is anchored in the development of standalone topics with distinct information types.</shortdesc>

<prolog><metadata><keywords><indexterm>topic-based authoring</indexterm></keywords></metadata></prolog>

<conbody>…</conbody></concept>

Example 2-- Insert block-level index tags immediately following the start tag

In the following example, you see two primary index terms, “DITA” and “topic-based authoring” and two secondary index terms, “topic-based authoring” and “introduction”. The indexterm element occurs immediately following the start tag of the block. The resulting index would read as follows:

DITA

topic-based authoring ……. 15

topic-based authoring

introduction ……. 15

<indexterm>DITA<indexterm>topic-based authoring</indexterm></indexterm

><indexterm>topic-based authoring<indexterm>introduction</indexterm

></indexterm>Topic-based authoring has been a mainstay of technical information development since we first began developing help systems. We learned quickly enough that we couldn't split our existing books into help topics by making every heading level a new topic. Information originally designed with a unique narrative flow no longer made sense nor assisted users in finding exactly the content they needed. We had to rethink the types of information that our help systems should include and create a new set of standards for their development. The result is topic-based authoring.

Example 2a—special case of example 2

In the following example, the index term that describes the section is placed immediately after the section title but before the text of the section. The index term will refer to the entire section. If the section starts a new page, the index page reference will correctly reflect the beginning page of the section.

<section><title> Reference Elements</title><indexterm>reference elements</indexterm>… </section>

Example 3—Insert a sentence-level index tag immediately preceding the sentence in which the text occurs.

In the following example, you see index terms included in a paragraph. The index terms refer to the final sentence in the paragraph and are placed immediately before that sentence. The resulting index would read as follows:

DTD

declaration assignment …….. 20

reference information type

adding

DITA reference topics include data that must be available to users for rapid lookup. The information in the reference topics may be managed by technical experts in databases so that it can be frequently updated, ensuring that the information in the reference topic is accurate.

<indexterm>XML declaration<indexterm>assigning</indexterm></indexterm>

<indexterm>DTD<indexterm>declaration assignment</indexterm></indexterm>

<indexterm>root elements<indexterm>adding</indexterm></indexterm>

<indexterm>reference information type

<indexterm>adding

<indexterm>root element</indexterm>

</indexterm>

</indexterm>Creating a DITA reference topic requires that you assign an XML declaration and that you begin with the root reference element.

Example 4—Insert word or phrase-level index tag immediately preceding the text to which it applies.

In the following example, the index term points to a specific term in the text. Remember that an indexterm element used in the middle of a sentence should not be relocated to avoid a fuzzy match with the segment in the TM. In the example, you must retype the words “properties table” within the indexterm element to create the index term. You cannot simply select the words “properties table” in the paragraph and turn it into an indexterm element.

Be certain to place the indexterm element immediately preceding the relevant term so that it is not orphaned if the term is removed or changed.

<dd>Use the properties element to create a <indexterm>properties table</indexterm> properties table to list all the details the user must know for the corresponding task topic. The properties element is the container for all the individual property elements your reference contains.</dd>

Example 5—Insert <index-sort-as> tags to accommodate translation sort-order changes.

The DITA 1.1 language specification explains the <index-sort-as> tag as follows:

Certain languages may have special sort order needs. For example, Japanese index entries might be written partially or wholly in kanji but need to be sorted in phonetic order according to their hiragana/katakana rendition. There is no reliable automated way to map written to phonetic text: for kanji text, there can be multiple phonetic possibilities depending on the context. The only way to correctly sort Japanese index entries is to keep the phonetic counterparts with the written forms. The phonetic text is presented as the sort order text for indexing purposes.

While the < index-sort-as > markup allows you to specify different sort orders for different instances of the same index entry, adding the < index-sort-as > markup for every index entry can lead to inconsistent sort orders in cases where you want the same sort order. You may specify a global default sort order by placing an indexterm instance in the map metadata (map/topicmeta/keywords/indexterm). There, the indexterm has no content context and will not generate an index page reference. Using <index-sort-as> in the map metadata will set the global sort order expression for that term. The individual indexterm instances in the content may override this global default with their own <index-sort-as> elements.

This is an example of an index entry for <data> that will be sorted as "data":

<indexterm><data><index-sort-as>data</index-sort-as></indexterm>

The map metadata entries must be prepared by the translator in conjunction with the localization coordinator to ensure that the sort order renders correctly.

Note the sort order must be modified for Japanese, Croatian, Adzerbadjiani, Serbian, Albanian, Mongolian, Georgian, and Caucasian. Consult with your localization service provider in advance to ensure that a mechanism exists for translators to create the sort-order terms in the DITA maps.

It may also be possible to modify the sort order of the index entries using a post-processing tool that is language specific.