



Indexing Effectively in DITA



Julio Vazquez, SDI Global Solutions

DITA is useful for helping writers create small units of organized information that can be used in multiple contexts. Of course, the readers' problem then becomes locating the information they want in a quick, reasonable timeframe. Although DITA provides enough metadata to simplify searching or even to present information the reader needs based on a profile, there are some media that cannot make use of those facilities. To bridge that gap, you can use the tried and true index.

The DITA elements you use to mark up the content are `<indexterm>`, `<index-see>`, and `<index-see-also>`. What makes DITA unique is where these elements can be placed and how the language specification states they should be processed. Depending on your localization, portability, and consistency requirements, you can use any of these placements to achieve your indexing goals.

Let's start with a quick look at the elements and how they should be processed and then we'll look at the correct placement within the content.

<INDEXTERM>

This element is the basis of indexing in DITA. It can contain text, other `<indexterm>` elements, `<index-see>`, or `<index-see-also>`. The rules and results for `<index-see>` and `<index-see-also>`, I'll cover in the discussion of those elements. Let's just look at the `<indexterm>` element and what the specification says about it.

When `<indexterm>` contains just text, the result is a primary entry that resolves to either the topic title or to a spot within the content; the entry can resolve to different things because you can place `<indexterm>` and its children in a number of places in your content source. An example of this form of the `<indexterm>` element is simply `<indexterm>indexable words</indexterm>`. I'll talk about placement later in this article.

By nesting `<indexterm>` elements within each other, you create secondary and tertiary index entries. The reference is located on the innermost term with no spot reference on the higher terms unless those were specified elsewhere. As an example, assuming no other index entries result in spot references at the primary or secondary level, when you code this:

```
<indexterm>recipes
  <indexterm>eggs
    <indexterm>benedict</indexterm>
  </indexterm>
</indexterm>
```

The transform generates an index entry that looks like this:

recipes
eggs
benedict, 10

<INDEX-SEE>

Use this element when you want the reader to refer to index entries that are synonymous with this specific term and that point to salient information. The parent of the `<index-see>` element cannot have any other children. An example of `<index-see>` looks like:

```
<indexterm>six-string
  <index-see>guitars, types of
</index-see>
</indexterm>
```

This coding results in an index entry that looks like this:

six-string see guitars, types of

Notice that there is no page reference; the index entry to which you direct the reader contains that information.

<INDEX-SEE-ALSO>

This element directs the reader to an entry for a similar object that has information that applies to both the current entry and the entry under the target of the see-also reference. The rule for `<index-see-also>` is the same as for `<index-see>`—the parent element can have no other children. The result is different in that the parent does resolve to a spot reference. An example of this sort of coding looks like this:

```
<indexterm>12-string maintenance
  <index-see-also>guitar maintenance
</index-see-also>
</indexterm>
```

This coding results in a structure that looks like:

12-string maintenance, 28

see also **guitar maintenance**

Elsewhere in the index is an entry for guitar maintenance that points the reader to additional information they might be interested in.

Another benefit of using `<index-see-also>` is that it helps you avoid secondary and tertiary entries in multiple index entries for the same information. Consider it a single-sourcing for indexes.

WHERE TO INDEX

DITA allows you to mark index terms within the content stream, in the prolog of a topic, or within the metadata of a `<topicref>` element within a DITAMAP.

When you place index terms within the content of a topic, the reference resolves to a spot reference to that specific point within the topic. This type of reference is useful only in extremely long topics or in the print realm. In general, long topics are discouraged when writing in DITA so this type of indexing is discouraged. However, there are good reasons for indexing this way:

- ◆ The primary output type is a printed document.
- ◆ The topic is, of necessity, long.
- ◆ The topic is a reference topic with many possible points of interest.

A problem with placing index terms within the content is that if the content is used in a different information set, the embedded terms can cause inconsistencies with existing index entries in that information set. You might need to spend time managing the index entries as you move the topic from one set of information to another.

When you place index terms within the prolog of a topic or in the metadata of the topic, the reference resolves into a pointer to the title of the topic. This is optimal for an online environment but could be a little less satisfactory if the topics are reused to produce printed output.

So why use one method over the other? The answer to this question depends on what your overall objectives are and what markup meets those needs. You need to consider localization, portability, and consistency. Your goals in each of these areas determine how you finally decide to index the information.

For maximum portability, consistency, and minimization of localization costs, I recommend that you remove the index terms from anywhere within the topic boundaries; place the index terms within the metadata of the `<topicref>` element that embeds the topic. Banishing index entries within the topic prevents the topic from introducing inconsistencies in a potential new information set and allows you to use the existing information set's index as a guide to the most appropriate entries for the topic's content. This method also makes the topic maximally portable.

How do we minimize the localization costs of indexing? You can meet that goal by leveraging the reuse capabilities inherent in DITA or in DITA 1.2 using the new keyref mechanism. As I am most familiar with the reuse function, I'll address that but will not discuss keyref in this article because the specification is incomplete. However, keep keyref in mind as it is designed to allow you to define referenced objects in the DITAMAP. If you want more information about the DITA 1.2 keyref features, see the Keyref Overview–DITA 1.2 at <http://dita.xml.org/resource/keyref-overview-dita-12>.

REUSABLE INDEX TERMS

If you have a number of information sets that have the same basic structure and use the same terms, you might notice that the index structure is the same or similar. Much like content that is reused in different information sets, you can use index structures in different information sets.

In keeping with maximizing the portability of the topics, put the index terms within the metadata for the `<topicref>` elements for each topic or in the `<topicmeta>` element before any `<topicref>` elements. To minimize the localization costs, use a container, an index-only DITAMAP, to hold the index terms, and `conref` those structures into the DITAMAP that you are using to create your output. The DITAMAP that contains the index terms never gets processed; its only reason to exist is as a source for reusable index terms. Structured this way, you translate only the index-only DITAMAP; the referring DITAMAPs do not contain any index content.

First let's look at the index-only (non-processed) DITAMAP. Let's call it `index_source.ditamap`, which has the index entries in a single location.

```
<map>
  <topicmeta>
    <keywords>
      <indexterm id= "about_maps_idx">maps
        <indexterm>description of
      </indexterm>
    </indexterm>
    <indexterm id= "dita_root_idx">DITA</indexterm>
    <indexterm id= "maps_strct_idx">maps
      <indexterm>structure of
    </indexterm>
  </indexterm>
  <indexterm id= "dita_concepts_idx">DITA
    <indexterm>concepts</indexterm>
  </indexterm>
</keywords>
</topicmeta>
</map>
```

It's now possible to reuse the <indexterm> elements in other maps using conref attributes. You cannot reuse the map-level elements within the body of the topic. Let's look at a DITAMAP (output_produce.ditamap) that uses this index-only map for its index entries to see how it refers to the index terms.

```
<?xml version= "1.0" encoding= "UTF-8"?>
<!DOCTYPE bookmap PUBLIC "-//OASIS//DTD DITA
BookMap//EN"
  "..\..\dita\dtd\bookmap.dtd">

<bookmap xml:lang= "en-us">
  <booktitle>
    <mainbooktitle>Practical DITA
    </mainbooktitle>
  </booktitle>
  . . .
  <chapter href= "global.dita">
    <topicref href= "basic_info.dita">
      <topicmeta>
        <keywords>
          <indexterm conref= "index_source.ditamap/
dita_concepts_idx"/>
        </keywords>
      </topicmeta>
    </topicref>
    <topicref href= "map_over.dita">
      <topicmeta>
        <keywords>
          <indexterm conref= "index_source.ditamap/
about_maps_idx"/>
        </keywords>
      </topicmeta>
    <topicref href= "map_struct.dita">
      <topicmeta>
        <keywords>
          <indexterm conref= "index_source.ditamap/
maps_strct_idx"/>
        </keywords>
      </topicmeta>
    </topicref>
  </chapter>
  . . .

  <backmatter>
    <booklists>
      <indexlist/>
    </booklists>
  </backmatter>
</bookmap>
```

When you process output_produce.ditamap through the DITA Open Toolkit, the index in the backmatter of the book contains the entries reused from index_source.ditamap merged into any other index entries either explicitly coded or reused from other sources.

KEY POINTS ABOUT A CENTRAL INDEX DITAMAP

Keep the following things in mind if you use a DITAMAP as a central index:

- ◆ Document extensively—make sure you keep track of what information sets are making use of the file both in the central index DITAMAP and in the DITAMAPs that are using that file.
- ◆ Manage the central index DITAMAP tightly—you still wind up with indexing problems if you don't somehow keep control of it. Whether a single person or a limited committee updates the file, that level of control is far better than allowing every member of the team the ability to update the DITAMAP.
- ◆ Make sure that you supply the index file to groups that reuse topics using that file—although the context in which the topic appears may not require the specific index entries in the central index DITAMAP, that courtesy can make indexing less of a chore for the reusing group.

SUMMARY

In this article, I have described three methods of placing index terms, which helps readers retrieve information using a familiar paradigm. I have also described how to leverage the reuse capabilities of DITA to ensure the consistency of index terms throughout an information set and reduce your localization costs by providing a single source for all your index entries. Which method you choose to use depends on your production and localization needs. That said, I highly recommend you use a central DITAMAP to contain index entries for your project, if you want to

- ◆ minimize your translation costs
- ◆ improve consistency
- ◆ standardize your index entries
- ◆ manage index entries easily

For a summary of the best practices for indexing DITA topics for translation, see <http://www.oasis-open.org/committees/download.php/27581/IndexingBestPracticesWhitePaper.pdf>. 