The Center for Information-Development Management
710 Kipling Street • Suite 400 • Denver, CO 80215 • 303/232-7586 • www.infomanagementcenter.com

ARTICLE REPRINT

Mapping out a DITA Map Comparator

Michael Anthony Smith, DeltaXML

INTRODUCTION

This article is based on our DITA Europe 2013 presentation[1] where we aimed to present the issues you should consider when performing DITA map comparisons so that the right form of map comparison can be selected for a given context. We deliberately keep most of the content at a high level, so it can be understood by DITA authors and managers. Those paragraphs that are intended for DITA developers are prefixed with a **"Technical Aside"** label.

For the purposes of discussion, we have taken the view that a DITA document is an ordered collection of topics and other resources, such as images, that are organised by a DITA map hierarchy. One reason for performing a DITA map comparison is to perform a DITA document comparison. In this case a map and its submaps can be viewed as a mechanism for ordering the topics within a DITA document. We shall refer to this as the "topic-centric" viewpoint.

There are other reasonable viewpoints to consider, including the "XML content" viewpoint, where the XML content of the maps is compared. This viewpoint may be appropriate for use in a Content Management System (CMS). Such alternative viewpoints are briefly considered at the end of the article and are the subject of ongoing and future work.

TOPIC-CENTRIC VIEWPOINT

There are a number of issues to consider for a topic-centric map comparison design, including:

- When in the document management workflow should the comparison be performed?
- What should be compared?
- How should topics and other resources be 'aligned'?
- What should the result of the comparison look like?

Before we get into these details, the first question to be considered is why the comparison is being performed as this sets the context for answering the other questions. For example, comparisons could be performed for

- **Review.** Enable a reviewer to view, accept, and reject changes. Review could use an editor-specific tracked change format.
- **Publication.** Show end users what has changed between versions. Publishing could use the DITA markup format. Here the expectation is that the publication process will suitably highlight changes marked up using DITA's standard revision attributes.
- **Archiving.** Enable a history of document change to be efficiently stored. Archiving could use a lossless patch format, which need not be human readable.
- **Auditing.** Provide traceability for legal, security, safety, or other domain specific purposes. Auditing could use a custom, possibly machine-processable markup for correlation with other non-DITA data.

Each of these goals leads to different requirements in terms of what changes are shown and how they are recorded. The great advantage of DITA, being in XML, is that we can potentially meet all these needs by different processing models and different output formats.

When to Compare?

At what stage in the document management workflow should the comparison be performed—At the beginning, the end, or somewhere in the middle? The obvious answer to this question is to do it right at the beginning and compare the DITA source—but is this appropriate? It probably is in the context of 'round trip' processing where the output is potentially going to be used as the new input, which could happen as part of a 'review' process. In that situation, we want to see what has changed in the DITA source in order to check that it is as we intended, but it could also be published as illustrated in Figure 1.

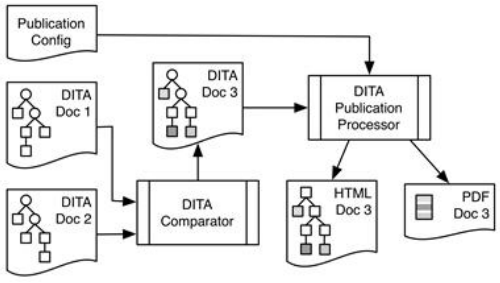


FIGURE 1: DITA SOURCE COMPARISON AND PUBLICATION

1 Mapping out a DITA Map Comparator, Michael Anthony Smith and Tristan Mitchell, DITA Europe 2013, http://www.infomanagementcenter.com/DITAeurope/2013/abstracts.htm#Smith

It is, however, not clear that source comparison is the best approach in other contexts, such as the publication context. From a publication perspective, it is not sufficient to know that the DITA topic's source has changed, because those changes may be filtered out by conditional processing. Further, knowing that a topic's source is unchanged does not guarantee that the associated publication output will be unchanged (see technical aside for details). Therefore, in the publication context, it may be more appropriate to perform a comparison after some pre-processing has occurred, for instance, after DITA's conditional filtering, key resolution, and content reuse mechanisms have been applied. Preprocessing reduces the complexity of the document structure and ensures that only the content that is relevant to the output is being compared, as shown in Figure 2.

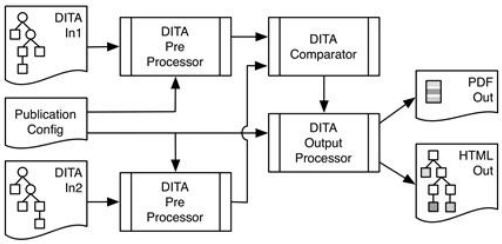


Figure 2: DITA Pre-processed, Compared, and Published

Technical Aside: Consider the comparison of a DITA document where the only change between two versions of a DITA document, D, is in the content of an element, E, in a resource only topic, RT, that is `conkeyref` included into D. In this case, a DITA map comparison on the source of document D will have a single changed topic, the resource only topic RT. It is possible to mark the `conkeyref` elements that refer to E as containing references to changed content (changed referents) by setting a DITA rev attribute to 'referent changed'. Note that this process requires the `conkeyref`'s key to be resolved.

Another alternative is to compare the outputs following publication, as illustrated in Figure 3. In this case, a comparator for each output format is required. Further, those comparators may have a more complex task as there is likely to be layout specific content in the files that may interfere with the content.

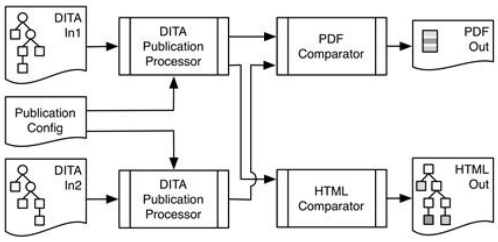


Figure 3: Compare Published Documents

What to Compare?

It may appear to be a little strange that we are questioning what should be compared. Is it not obvious that we are comparing two DITA maps? Certainly, but what is a DITA map? It is an XML file that is used to specify how topics are assembled to produce a document. Here, the nested hierarchy of topic element references is used to define the primary content of a map. Therefore, a map comparison could be interpreted as comparing the elements that define how the topics are assembled. Such a viewpoint may be appropriate in the context of a content management system. Alternatively, it is possible to treat the map as if it included all the referenced topics (and other resources), which is what we are doing in the topic-centric part of this article.

Having said that a topic-centric comparison compares the content of referenced topics (and other resources, such as images), should it compare all the referenced resources? The simple answer is probably "no." Some of the referenced resources, such as an external web site, may not actually be part of the document; we do not necessarily want to compare, or provide, the means of comparing such resources. Fortunately, DITA provides the concept of a scope, which is applied to referenced resources. Scope can be used to specify whether a resource is part of the document (and thus included in the comparison).

Technical Aside: DITA provides three levels of scope: local, peer, and external. Here local and external scoped resources are likely to be included and excluded from the document respectively. Peer scoping is more problematic, as the precise meaning of peer scoping is context/implementation defined. Therefore, a comparator that is not aware of such context/implementation details could provide an option for treating peer-scoped resources as either local or external.

The answer to what to compare is context dependent, but from the topic-centric viewpoint, the context would consist of those resources that are part of the document.

Technical Aside: The comparison of non-DITA resources is likely to vary from comparator to comparator. It could be that non-DITA resources are not compared, are compared using a binary equality, or are compared in a format-specific manner. One of the issues with a format-specific comparison is how to represent the differences.

Topic/Resource Alignment

A key issue for a document comparison is aligning the content between the two documents being compared. Unfortunately, it is not quite as simple as it sounds, so we now focus on the topic alignment issue. Here each topic that is considered part of one of the input documents needs to be either aligned with a corresponding topic in the other document or identified as distinct, as it appears in only one document.

There are several approaches to this alignment problem:

- **Content-based Alignment.** Here the topics are aligned in terms of their content so that the topics most similar to each other are considered to be corresponding topics. Content-based alignment is a computationally expensive approach which is O(n2). This is because each topic that is part of one of the inputs has to be compared to every topic that is part of the other input.
- **Relative URI-based Alignment.** Here topics with the same relative locations in the top-level source map are considered to be the same. Note that topic URIs that cannot be made relative to their top-level map have to be kept absolute.
- **History-based Alignment.** Here the topics are aligned based on a version control history or content management identifier.
- **Pattern-based URI Alignment.** Here topics are aligned based on some supplied regular expression, catalog, or other rewrite rule system.
- **Multi-scheme Alignment.** Here the various alignment approaches are tried in some order, and the first (or possibly best) match is taken.

Once the topics are aligned, it is then possible to perform an XML content comparison on each pair of aligned topics. Such comparisons could be provided with additional DITA-map context information, such as the defined values of keys, and the content of any content reuse mechanism. Performing a quality comparison of DITA topics is complicated and beyond the scope of this article.

Technical Aside: Assuming that content reuse mechanisms have not been resolved and expanded prior to comparison, it may be worth performing a two-pass comparison process. The first pass is to compare all the topics before reused content is taken into account. The second pass is to adjust the results to take the reused content into account. This separation of concerns allows the topic comparisons in both passes to be performed concurrently.

Output Formats and Markup

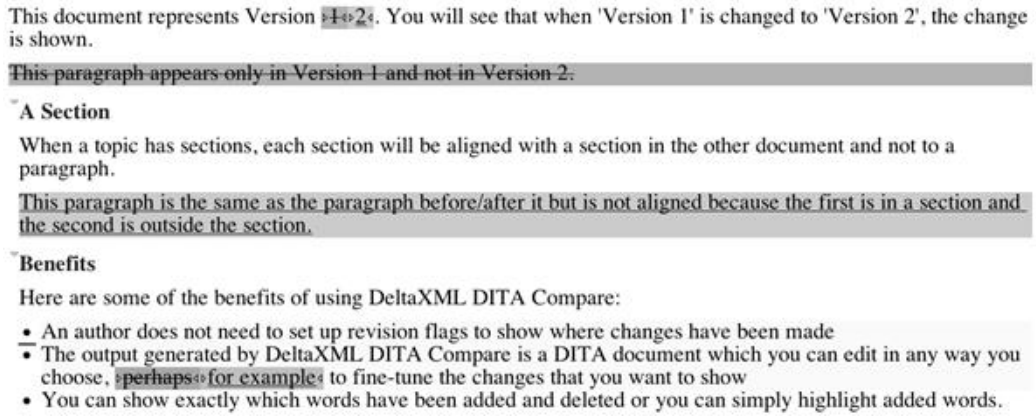
Once the DITA topics have been aligned and compared, the differences need to be marked up in a suitable output format:

- **DITA Markup Format.** Here the differences are marked up using DITA's rev and status attributes, which can be used by a standard DITA publishing pipeline. Figure 4 illustrates an XML editor's view of such markup.
- **Tracked Changes Format.** Here the differences are marked up in an editor-specific tracked change format, which allows the changes to be reviewed and accepted or rejected.
- **Patch Format.** Here the differences are marked up using a complete and accurate markup language, which is intended to allow an input to be created from the other input and the patch file. One use case is to store a versioned history as a sequence of patch files.

The change-marked topics then need to be gathered together into one or more maps, which ought to specify which of their referenced topics has changed. Here, the simplest approach is to produce a flat map that contains a topic reference for each of the topics, which is labelled with its status (inserted, deleted, changed, or unchanged). We refer to such an output as a "topic-set result."

Technical Aside: Using DITA's `status` attribute to specify the status of what a topic reference is pointing to (rather than the status of the topic reference itself) may not be appropriate. However, repurposing the typical use of the `status` attribute in this case appears to provide useful information in standard DITA format at little cost. An alternative could be to introduce either custom processing instructions or a DITA specialization.

The straightforward topic set result enables the changes in the topics to be identified and reviewed, but is not suitable for onward publication, as the structure of the document has been lost. Further, it explicitly uses DITA Markup as its map-level output format, which may not be appropriate. However, some of the other topic-level output formats, such as the editor tracked


This document represents Version 1 2. You will see that when 'Version 1' is changed to 'Version 2', the change is shown.

~~This paragraph appears only in Version 1 and not in Version 2.~~

A Section

When a topic has sections, each section will be aligned with a section in the other document and not to a paragraph.

This paragraph is the same as the paragraph before/after it but is not aligned because the first is in a section and the second is outside the section.

Benefits

Here are some of the benefits of using DeltaXML DITA Compare:

- An author does not need to set up revision flags to show where changes have been made
- The output generated by DeltaXML DITA Compare is a DITA document which you can edit in any way you choose, ~~perhaps~~ for example to fine-tune the changes that you want to show
- You can show exactly which words have been added and deleted or you can simply highlight added words.

FIGURE 4: DITA MARKUP TOPIC COMPARISON RESULT

change formats, tend not work well at the map level, as there is often no way of representing the modification of an attribute.

Technical Aside: Representing a change in a `topicref` element's `href` attribute could be done by deleting the original `topicref` element and adding the new `topicref` element, but any nested `topicref` elements would also be added and deleted.

Retaining the hierarchical topic structure of both input maps in the comparison result is, in general, not possible as these hierarchies can have conflicting structures. However, it is possible to retain one of the input topic hierarchies in the result. An obvious choice is to retain the hierarchy of the updated document. In this case, the topic references within the retained hierarchies can be marked with the status of their topics as previously discussed. The remaining issue is what to do with those topics that do not appear in the chosen hierarchy. The simplest approach is to gather these remaining topics into a new map. Thus, the result of the comparison is a pair of maps, one containing an updated hierarchy, and the other the remaining topics. We refer to such an output as a "map-pair result." Figure 5 illustrates an XML editor's view of such a result, where the two maps have been added into a master map.

So far we have presented two types of map-level comparison result structures, the topic-set and map-pair. It is possible to construct several other map-level result structures. For example, a variant of the map-pair result where the remaining (typically deleted) topics are inserted into the updated map at 'appropriate' points. Here, the challenge is working out where the 'appropriate' point to insert each remaining topic is. Such an algorithm would have to take into account that the updated structure may have moved the relative locations of the topics in the original structure.

Alternative Viewpoints

So far we have discussed the topic-centric viewpoint in some depth, as this mirrors the focus of our thoughts on DITA map comparison. However, there are other potential viewpoints for a DITA map comparison:

- **Monolithic viewpoint.** Here the map is used to construct one linear source document, which is then compared using an XML comparator. Having done this, it would be possible to split such a document back into its constituent parts. This approach makes sense in situations where the boundaries between constituent parts 'flow,' for instance, the content from one part may flow into a neighbouring part due to the insertion of some text or structure.
- **XML content viewpoint.** Here the map is considered to be just another XML document, and we are interested in the changes between two versions of it. This viewpoint may be appropriate in a Content Management System (CMS) context.

Technical Aside: It is easy to state that within a CMS context we want to compare the XML content of the map itself. There are, however, practical issues of producing a reasonable comparison output. For example, is it appropriate to only allow `topicref` elements to align when, and only when, their href attribute values are the same? It might be better to perform a hierarchical alignment of the `topicref` elements. Note that in this case, the conflicting hierarchies issue, which we have already discussed, is likely to be encountered.

DITA Map Comparison in Practice

Version 5.0 of our DeltaXML DITA Compare tool[2] provides topic-centric DITA map comparison support, along the lines discussed in this article. It also provides a DITA Open Toolkit[3] customisation, which assists in representing change in a PDF publication[4]. Future versions of the DITA Compare product will include specialist map-comparison support for CMS vendors, provide additional comparison output formats, and provide improved support for the DITA publication process. We are always interested in discovering other DITA map comparison contexts and welcome feedback and suggestions.

Summary

DITA map comparisons can be performed for a variety of purposes, including source review, patching, and publication. These purposes provide the context for selecting what should be compared, when it should be compared, how to perform the comparison, and how to present the results. Overall, we conclude that there are several forms of DITA map comparison that are both useful and feasible to perform, each for its intended purpose.

2 DeltaXML DITA Compare, http://www.deltaxml.com/products/dita/
3 DITA Open Toolkit, http://dita-ot.github.io
4 Guide to Publishing with DITA Open Toolkit, http://www.deltaxml.com/products/dita/samples/publishing-with-dita-ot/

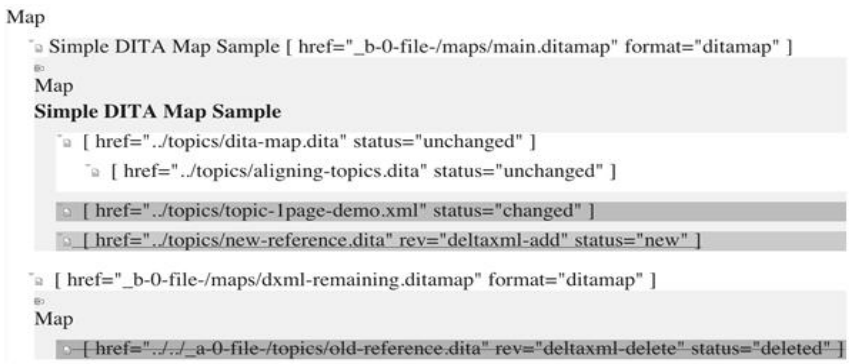


Figure 5: Map-Pair Result