

Background

In March of this year, the document, Steve Manning authored the document, Aggregated Authoring: Need and Functional Description, and posted it to the OASIS website on behalf of the OASIS DITA for Enterprise Business Documents Subcommittee (DITA BusDocs). The purpose of the document was to describe and clarify the need for aggregated authoring for tool vendors and other interested parties. The document was not intended to be prescriptive regarding technical approaches, but did acknowledge that, without modification to the DITA specification, there were two potential approaches:

- Edit a number of topics within a single file using the current architecture that DITA provides for this purpose, with the ability to transform this single file to and from individual topics and a related map.
- Acknowledge that some authoring tool vendors might support authoring against both the map and topic DTDs at the same time, using proprietary technology to expand the topics within map for authoring, while saving and loading individual topics and a related map.

In the six months since the original Aggregated Authoring document was posted, the BusDocs subcommittee has had numerous discussions with other DITA technical committee members, DITA practitioners, and end users. The purpose of this document is to build on the knowledge gained in order to provide a “suggested implementation” based on the first approach described above which leverages the current DITA standard. While the subcommittee recognizes that tool vendors will always be able to provide proprietary solutions at will, we believe that offering an off-the-shelf recommendation based on the standard will allow the broadest tool support and simplest implementation of DITA for the widest possible range of enterprise business documents.

The subcommittee has incorporated excerpts from the Darwin Information Typing Architecture (DITA) Version 1.2 Committee Draft 03 dated 22 June 2010. Readers are encouraged to consult the current version of that document as the authoritative source for referenced content.

Audience and Purpose

This document has several potential audiences and associated purposes:

Audience	Purpose
DITA Technical Committee members	The BusDocs subcommittee is requesting that the DITA Technical Committee vet the approach recommended in this document and consider providing it the appropriate “official” status ¹ .
DITA Practitioners	A growing number of DITA practitioners have implemented a range of business document types with aggregated topics. Feedback regarding the similarity of this document to their current approach and suggestions for improvements to the design are welcomed.
Tool Experts and Vendors	To the best of our knowledge, the approach in this

¹ This document defers to the Technical Committee not only the viability and final design and of the approach, but the best way to incorporate it into the standard’s various specializations, recommendations, etc.

	document can be implemented in a standard XML editor without requiring code changes or extensive configuration. Tool experts and vendors are asked to review the document to ensure that this goal has been met.
End-user Business and Technical Strategists	Strategists who are researching the correct approach to an enterprise authoring schema, or examining DITA's applicability to enterprise business documents, are asked to provide feedback on document clarity and perceived strategic value of the approach.

In terms of technical expectations, we anticipate that all audiences are at least familiar with the basic terminology and architecture of the DITA standard. We suspect that potential adopters of this solution may have only a rudimentary understanding of DITA, and therefore we provide more information than would be required for Technical Committee members.

Introduction

The reader is encouraged to first read background information on DITA and the document [Aggregated Authoring: Need and Functional Description](#), unless already well acquainted with DITA and the rationale behind the DITA BusDocs subcommittee.

We define "enterprise business documents," as the complete taxonomy of narrative documents created by knowledge workers within a commercial or public organization. The term narrative documents is used to exclude those documents that would be better classified as "forms" or "tabular reports" that do not also contain significant amounts of narrative content.

Technical documentation is an example of a major branch within this enterprise taxonomy, and a vital part of the information exchange that organizations hope to gain when implementing DITA enterprise-wide. However, for the purpose of this document, the term *business documents* should be taken to exclude technical documentation, since the organizations we seek to help are very pleased with the information that has been provided to date for implementing DITA for technical publications, and/or are focusing on a set of documents that has different characteristics.

BusDocs Subcommittee Scope

While the BusDocs subcommittee is concerned with a very broad range of documents across most public and commercial industry segments, the depth of the BusDocs subcommittee's suggested involvement in these areas is limited to the subjects in this document and related papers. The subcommittee views the subjects of this document to be foundational issues that affect all business documents regardless of industry. Our concern is that failure to address these issues once with a global view will result in a variety of approaches being implemented in different subcommittees. This would be a waste of resources, a hindrance to interoperability, and a source of continual confusion to anyone who needs to implement the work of more than one of the growing number of business document oriented subcommittees.

Aggregated Topic Authoring

This section of the document is concerned with the issues of:

1. Authoring a number of topics as a single “document,”
2. Storage of the document as a map² and related topics,
3. A lossless interoperability between subsequent standalone and aggregated authoring sessions.

1. Authoring a Number of Topics as an “Aggregated Document”

The current DITA standard provides two off-the-shelf methods to create a single document that is composed of multiple topics. The DITA BusDocs subcommittee has seen both of these methods used in various projects and therefore offers some suggestions on the relative benefit of each approach that have led us to the current recommendation.

The basic DITA topic types allow nesting of topics within topics and therefore are capable of representing what we are calling an aggregated document. For this general nesting of topics within topics, DITA specifies two restrictions:

- A topic nested within a topic must be of the same type. For example, concept can contain another concept and task can contain another task.
- Nesting must occur after the close of the container topic body.

The DITA standard recognizes that there are cases where *heterogeneous* topic types need to be nested; and the ditabase definition (dita.dtd or dita.xsd) exists for this purpose. Ditabase allows <dita> to be used as the root element in place of a topic type, and in doing so expands the nesting capabilities of topic to include heterogeneous topic types. Ditabase does not remove the restriction that nesting must occur after the close of the topic body.

The following provides additional detail on the relationship between ditabase and topic nesting:

Ditabase allows topic, task, reference, glossentry, or glossgroup as peers in the root element:

Doctype	Content model
ditabase	(topic or concept or task or reference or glossentry or glossgroup) (one or more ³)

² The subcommittee is interested in defining an unambiguous model of how a large document might be saved as *multiple* maps and their related topics. Due to the architecture of maps and map references, we believe that this enhancement can be added at a later date. For the purpose of simplicity, this document will only address saving a single map and related topics.

³ Source of this and similarly formatted content: Darwin Information Typing Architecture (DITA) Version 1.2 Committee Draft 03 dated 22 June 2010

Example

```
<dita>
  <concept id="a">...</concept>
  <task id="b">...</task>
  <task id="c">...</task>
  <task id="d">...</task>
</dita>
```

DITA 1.2 Language Reference: <dita> Element Content Model

This creates a document that at its first level of nesting can contain all topic types defined within database. It can also accommodate all topic types at each subsequent level of nesting since it expands the topic content model to allow heterogeneous nesting: The following table shows the different content models for Concept when used with a root element of <concept> or a root element of <dita>. (Only these two cases are discussed because the restriction on heterogeneous nesting prohibits concept from being used with any of the other topic types as the root element.)

Doctype	Content model
Concept	((title) then (titlealts) (<i>optional</i>) then (abstract or shortdesc) (<i>optional</i>) then (prolog) (<i>optional</i>) then (conbody) (<i>optional</i>) then (related-links) (<i>optional</i>) then (concept) (<i>any number</i>))
Ditabase	((title) then (titlealts) (<i>optional</i>) then (abstract or shortdesc) (<i>optional</i>) then (prolog) (<i>optional</i>) then (conbody) (<i>optional</i>) then (related-links) (<i>optional</i>) then (topic or concept or task or reference or glossentry or glossgroup) (<i>any number</i>))

Expansion of the Concept Content Model within Ditabase

Concerning the use of one or the other of these approaches in projects to date, it should be clear from this discussion that projects which did not require heterogeneous nesting of topic types were free to implement ditabase or a specific topic type as their root element for authoring. Projects that did require heterogeneous nesting of either existing or specialized topic types were implemented within ditabase since implementation within any topic type would not have supported the requirements.

The generic busdocs document model proposed in this series of papers does not address specific document types and therefore topic, concept, or perhaps other topic types could be successfully used as the root for aggregated authoring. However, our desire is to provide a model that will address the broadest range of use cases, which leads to a recommendation of ditabase. For those who might be concerned that use of topic without ditabase makes more sense in some cases, we are providing a discussion on perceived advantages/disadvantages of each of these two approaches.

Regardless of which root element is used (<dita> or <topic>) the basic architecture of the aggregated document is the same. We will look at this architecture first, using ditabase as the root, and then follow with the discussion of the differences between using dita or topic as the root element.

Example

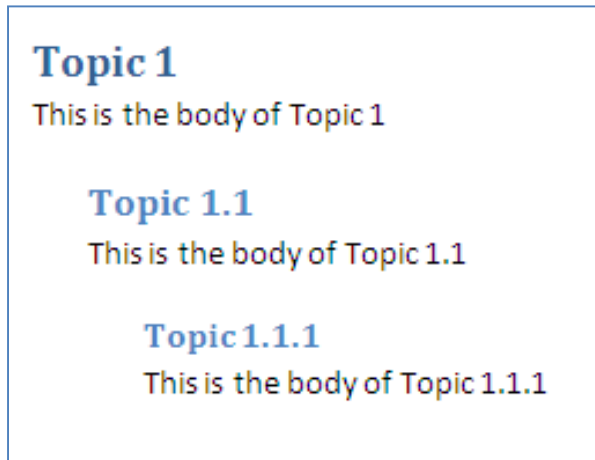
```
<dita>
  <concept id="concept">
    <title>Introduction to Bird Calling</title>
    <conbody>
      <p>Bird calling requires learning:</p>
      <ul>
        <li>Popular and classical bird songs</li>
        <li>How to whistle like a bird</li>
      </ul>
    </conbody>
    <task>
      <title>How to Whistle Like a Bird</title>
      <taskbody>... </taskbody>
    </task>
  </concept>
</dita>4
```

DITA 1.2 Language Reference: **<concept>** Element Content Model
Bold Text Highlights Expansion of Content Model under **<dita>**

⁴ Adapted from: Darwin Information Typing Architecture (DITA) Version 1.2 Committee Draft 03 dated 22 June 2010

2. Storage as a Map and Related Topics

This discussion begins with a simple document example:



The aggregated document form of this document appears as:

```
<dita>
  <topic id="topic 1">
    <title>Topic 1</title>
    <body>
      <p>This is the body of Topic 1</p>
    </body>
    <topic id="topic 1.1">
      <title>Topic 1.1</title>
      <body>
        <p>This is the body of Topic 1.1</p>
      </body>
      <topic id="topic 1.1.1">
        <title>Topic 1.1.1</title>
        <body>
          <p>This is the body of Topic 1.1.1</p>
        </body>
      </topic>
    </topic>
  </dita>
```

This document could be saved as a map and associated topics, assuming that each of these topics is saved with a filename that matches its title:

```
<map id="samplemap">
  <title>Aggregated Topics</title>
  <topicref href="topic 1.dita" type="topic">
    <topicref href="topic 1.1.dita" type="topic">
      <topicref href="topic 1.1.1.dita" type="topic" />
    </topicref>
  </topicref>
</map>
```

This simple example provides a basis to discuss more the more complete requirements for pairing an aggregated document with singly stored topics and an associated map. The approach must meet the following requirements:

1. The BusDocs Aggregated Authoring Solution must unambiguously identify map topicref and topichead elements in the aggregated document without requiring the author to edit attributes
2. The BusDocs Aggregated Authoring Solution must allow the author to change a topichead to a topic or a topic to a topichead in the simplest manner possible

The following example illustrates a common structure encountered in business documents. The difference between this and the simple example provided above is that titles are nested directly in some cases without the inclusion of content between a parent title and a child title:

Table 1: Use Case Example of Mapping of Topic to Topicref and Topichead

Business Document Content	Related Map Component	Determining Element
Part II: Terms and Conditions of Award	topichead	
Overview of Terms and Conditions of Award	topichead	
Program-Specific Terms and Conditions	topicref	
The terms and conditions of award... ...appropriate for the cooperative agreement.		body/content
Effect and Order of Precedence	topicref	
Any waivers of or deviations... ...take precedence over Part II of the HHS GPS.		body/content
Requirements under Contracts	topicref	
The terms and conditions...		body/content
Public Policy Requirements	topicref	
This section addresses...		
Standards of Conduct for Recipient Employees		

In the above example, Overview of Terms and Conditions of Award represents a topichead while Public Policy Requirements, which exists at the same level of heading in the authored document, represents a topicref. The clear difference between these is that one includes a body while the other does not, which naturally led to the idea that a topic containing a body could be mapped to topicref within a map while a topic that does not contain a body could be mapped to topichead. This fulfills the first requirement of an unambiguous mapping that does not rely on an author editing attributes.

Using the body element for this purpose also fulfills the second requirement of allowing the author to interchange topicref and topichead in a simple manner. Although this represents an element change in the map, for the aggregated document author it is a natural additional or removal of content that exists under a "heading" in the document. This allows the author to:

1. Write the document without concern for how it represents map components. It is likely in these use cases that the author may not be aware that maps exist or that they contain structural distinctions such as `topicref` and `topichead`
2. Change any topic representation from or to a `topicref` or `topichead` by simply adding or deleting content. For example, as soon as a body is added to Overview and Terms and Conditions of Award, it would change the representation from a `topichead` to a `topicref`.

The Choice between Authoring within DITA or Topic

The fact that multiple topic types can be nested within `<dita>` might seem to make the choice of database over topic obvious. However, various sources⁵ have indicated that the decision is not so straightforward. This is because:

- There are some perceived drawbacks to using database
- Nesting of homogeneous topic types is sufficient for many use cases
- If the local copies of the shell DTDs⁶ provided in the DITA package are modified for this purpose, nesting of heterogeneous topic types is possible with any topic type at the root of a document

The BusDocs subcommittee would summarize the pros and cons of these approaches as follows:

Approach	Pro	Con
database	<ul style="list-style-type: none"> • Requires no modification to shell DTDs or specialization to nest core DITA topic types • Meets a broader spectrum of use cases with the core DITA topic types • Meets all use cases with specialization of topic 	<ul style="list-style-type: none"> • Does not provide a way to represent the document title without using a topic for this purpose • Adds the seemingly unnecessary DITA element to the document which may confuse business authors
topic with OTS shell DTD	<ul style="list-style-type: none"> • Requires no modification to shell DTDs or specialization to nest homogeneous topic types • Supports the aggregated document model without the addition of a seemingly unnecessary DITA element 	<ul style="list-style-type: none"> • Only provides support for the limited number of use cases that can be modeled with a single topic type
topic with modified shell DTD	<ul style="list-style-type: none"> • Supports all core DITA topic types • Encourages local copies of the shell DTDs which is considered a “best practice” by some DITA technical committee members 	<ul style="list-style-type: none"> • Modifying the shell DTDs as a requirement of configurations for BusDocs adds complexity and potentially confusion

⁵ There are threads in DITA groups and various blog posts that discuss this. Do we actual physical references to these?

⁶ Need to add hyperlink to good explanation of shell DTDs.

The determination of which approach to recommend is an architectural decision that we expect to be vetted and determined at the Technical Committee level. The subcommittee would suggest that:

- Aggregated authoring of a generalized business document model can be accomplished off-the-shelf with no modifications with ditabase, topic, or concept, at the document root
- The use of ditabase over topic or concept is preferred since heterogeneous nesting significantly broadens the document types that can be modeled with DITA off-the-shelf
- It appears to this subcommittee that ditabase was originally designed to model legacy aggregated documents that contained heterogeneous topic types. Therefore the use of ditabase for BusDocs configurations is true to the original design of ditabase, if not the original purpose⁷
- Implementers will examine specific document types to determine if and how to specialize and when doing so can determine the appropriate changes to the OTS DITA package
- The BusDocs subcommittee intends to create a number of specializations to illustrate DITA's application to specific business document types. A sample approach to editing the OTS DITA package will be part of these specializations

⁷ Some members of the Technical Committee view the purpose of ditabase (**reference needed?**) to be limited to conversion only, where a legacy document is converted first to topics nested within the dita root, and then subsequently (perhaps with manual effort) to individual topic files and a map.

3. Interoperability between Standalone and Aggregated Authoring

Basic Approach

The following diagram shows the basic architecture of the BusDocs Aggregated Authoring Solution using:

1. Database as the root node allowing inclusion of nesting of topics and its various specializations
2. Topic unspecialized and unconstrained⁸
3. Pre-and post processing provided by the BusDocs Subcommittee deliverable to transform between the database document and a standard map and topics
4. Standalone authoring of the unconstrained map and unconstrained topics

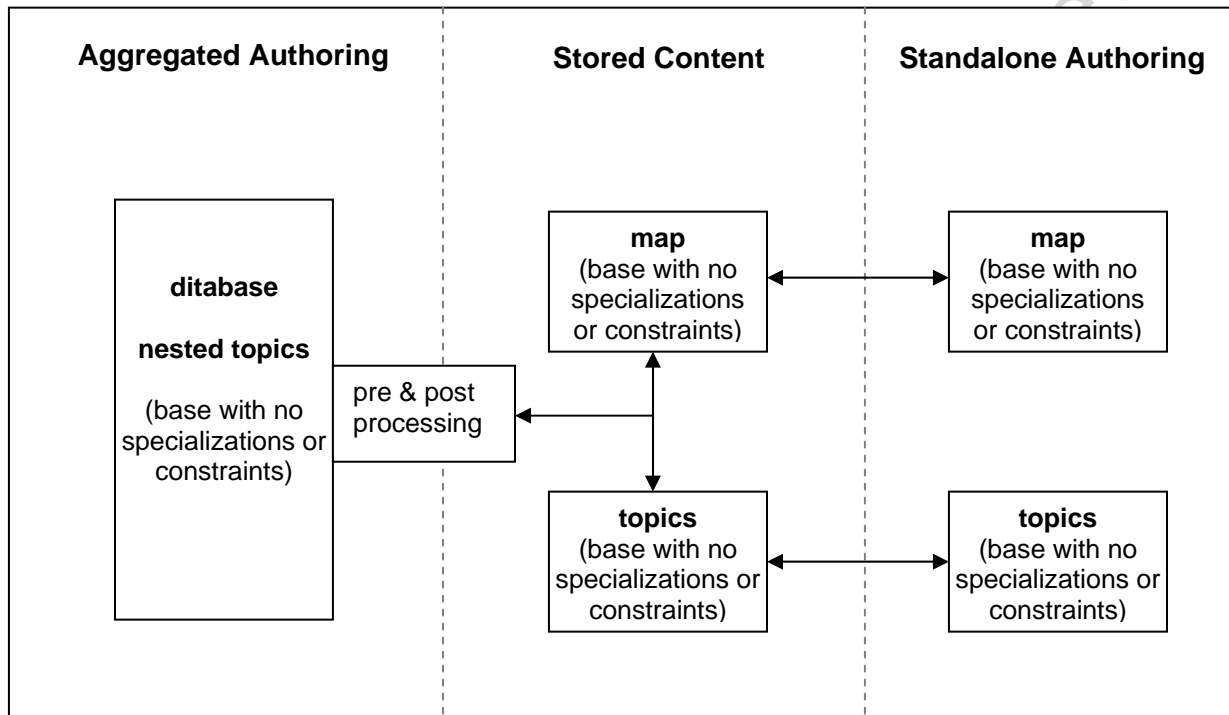


Figure 1: Basic BusDocs Aggregated Authoring Solution Architecture

⁸ Map and topic do not need to be specialized or constrained for the aggregated authoring approach. We expect that topic will be specialized to meet the needs of specific business documents, and that it may be constrained to create a “BusDocs base topic” designed to meet generalized business document requirements. The discussion of specializing or constraining topic is outside the scope of this document which deals only with aggregation issues.

Persistence of Map Content with Database Nested Topics

This discussion focuses on the requirement to persist map content when there is a “roundtrip” process of editing the map in an aggregated and standalone environment.

The simplified use case is:

1. A business user creates an aggregated document which is saved as a map and associated topics in a CMS.⁹
2. A content architect revises the map as an independent object using advanced map features not expected to be exposed to authors of aggregated business documents.
3. The business user revises the aggregated document without regard to the advanced map features used by the content architect.
4. The aggregated document is checked back into the CMS as an individual map and associated topics.

The associated requirements are that the BusDocs aggregated authoring configuration must:

1. be able to save an authoring session as one or more valid DITA maps and associated DITA topics.
2. be able to instantiate an authoring session by loading and assembling a valid DITA map and its references.
3. provide an unlimited number of lossless roundtrips of a map that represents nested topics within database
4. provide standalone authoring of maps that supports as much native map functionality as possible
5. provide aggregated authoring of topics that is as unencumbered as possible by the persistence of structure required only for standalone editing of maps or topics
6. not impose the requirement that aggregated authors edit any map directly
7. support editing of an aggregated document with the same CMS version control and file locking functionality for the aggregated map and topics that would be provided if the map and topics were edited standalone.
8. provides “round-tripping” of a map between aggregated and standalone authoring while persisting all map content that is not purposely modified in the aggregated authoring session

⁹ The BusDocs subcommittee views CMS storage as the norm for enterprise business documents. For this reason, we have incorporated a generalized model of CMS integration based on the ability of a CMS to:

- Store topics and maps as separate objects
- Provide check-out of an unlimited number of topics and a map into local storage for editing

While we suspect that aggregation of topics into a single document based on a map would be a desirable feature, we leave that to individual vendors and implementers. The configuration as described in this document will provide interoperability between topics and maps stored in the file system and an aggregated document.

General Approach to Support for Persistence of Map Content

As expressed above, enterprise requirements will demand that the standalone map editor expose map features that are not required for aggregated authoring users. Since usability and simplicity is critical to the aggregated authoring audience, a basic premise of the aggregated authoring configuration is that the tag set must be as sparse as possible. This creates a conflict between the standalone map authoring requirements and the aggregated document authoring requirements.

The BusDocs subcommittee has been discussing this issue for some time and previously considered what may seem to be the most obvious solution, but one that we found did not meet all of the above requirements. This solution is covered here for the benefit of readers who may consider it the first choice and wonder why it has not been adopted.

The aggregation of topics within database is lossless, since the content models for aggregated and standalone topics are the same. However, map and topic represent different structures with different purposes and the content model for map is substantially different than the content model of database or topic. This leads to issue of how this additional map content is preserved through multiple iterations of standalone and aggregated authoring.

The first solution considered was to specialize topic to hold and persist the additional map content. This immediately led to a conflict with requirements to keep the aggregated authoring content model as simple as possible for non-technical business document authors. Therefore this solution went further to constrain map to only include the most necessary features thereby limiting the content that would need to be persisted in the aggregated document.

This approach emphasized aggregated authoring simplicity over standalone map functionality. In order to recommend the approach the BusDocs committee would need to determine that:

1. Generally, aggregated business document use cases will not emphasize topic reuse or conditional publishing, and therefore advanced standalone map editing features will not be required. This assumes that for aggregated business documents, the authored form of the document will most often be the only form, and a simple map will suffice.
2. In those use cases where aggregated content is to be repurposed and conditionally processed based on robust map features, one or more secondary maps could likely be used for this purpose. Aggregated document authors would never see this secondary map.

While the above view (summarized as “aggregated business documents are different”) has some substance, we will not really know how important it will be to support specific advanced features of map until we receive feedback from real world implementations. Organizations may require more reuse and conditional processing for aggregated business documents than expected, and the use of secondary maps for this purpose may prove impractical. The subcommittee would prefer a solution to the apparent conflicting requirements that will not force us to revisit the architecture later if some of the members’ assumptions about standalone map requirements prove wrong.

For this reason the subcommittee has included strong standalone map functionality in the requirements for the configuration and favors an alternate approach that does not constrain maps but instead utilizes pre and post aggregated authoring transforms that BusDocs subcommittee intends to provide as a deliverable. This approach is further guided by the following assumptions made by the subcommittee:

1. Database cannot be specialized. Any persistence mechanism required only at the root of map would require an additional specialization of topic for this purpose
2. Adding a domain specialization for the persistence of map attributes within topic and its derived information types would not solve the aggregated authoring usability issues

3. Mixing elements or attributes required for roundtrip support in a structural specialization of topic is an improper architectural approach

Based on these assumptions, the aggregated authoring solution addresses the conflict between complex map publishing requirements and simple aggregated authoring by duplicating the map *structure* in ditabase using topic as the primary building block and using ids for referential integrity. There is no requirement to duplicate the map *content* in the aggregated document, which is clear from the following process:

1. Aggregated document is created and saved as a map and topics
2. Map is edited as a standalone entity and extended with a number of elements and attributes
3. Map and topics are aggregated once again for authoring. Content of the map is not transformed into the aggregated document since there is no requirement for this content in the aggregated authoring use case.
4. The save process incorporates a BusDocs subcommittee deliverable (minimally a transform but potentially an executable and transform) that saves the modified topics and:
 - a. Inserts new topicrefs/topicheads into the map
 - b. Deletes any topicrefs/topicheads and associated content that were deleted from the map during the authoring session
 - c. Changes the location of some topicrefs/topicheads and associated content within the map. This is handled by the aggregated authoring solution transforms utilizing the topic ids for referential integrity.

Structural Reconciliation of Map with Aggregated Authoring

The structural reconciliation of map with aggregated authoring occurs on two levels:

- At the root level, map is reconciled with ditabase.
- Since ditabase is a non-recursive element, map elements that exist below the root level will need to be reconciled with topic.

The following is a list of the elements in map and their relationship to the aggregated authoring document.¹⁰

Table 2: Reconciliation of Map with Topics Nested in Ditabase

Elements in Map (base)	Relationship to Topics Nested in Ditabase
(title) (optional)	Not replicated in the aggregated document
then (topicmeta) (optional)	Not replicated in the aggregated document

¹⁰ Elements that are not replicated in the aggregated document will persist in the map. The relationship between elements that are not replicated and map changes resulting from aggregated authoring needs to be examined in detail and added to this paper.

then any number of anchor	Not replicated in the aggregated document
or data	Not replicated in the aggregated document
or data-about	Not replicated in the aggregated document
or navref	Not replicated in the aggregated document
or reltable	Not replicated in the aggregated document
or topicref	Logically derived from topic (or its specializations) when the topic appears <i>with a body</i> in the aggregated document. Reference persisted with id attribute.
or anchorref	Not replicated in the aggregated document
or keydef	Not replicated in the aggregated document
or mapref	<p>Temporarily not supported.</p> <p>Logically derived from topic (or its specializations). Reference persisted with id attribute.</p> <p>A mechanism for this has not yet been identified. Support for the feature is constrained by the need for non-ambiguous transformation of topics into a map structure. It is <i>not</i> expected that aggregated authors will be aware of map hierarchies, so logic will need to be based on structural patterns as opposed to attributes provided by the aggregated author.</p>
or topicgroup	<p>Not replicated in the aggregated document</p> <p>There is some question as to the effect of new topics and topicref moves on topicgroup and whether the aggregated authoring solution transforms will be able to incorporate sufficient logic. Use cases need to be identified.</p>
or topichead	Logically derived from topic (or its specializations) <i>without a body</i> . Reference persisted with id attribute.
or topicset	<p>Not replicated in the aggregated document</p> <p>There is some question as to the effect of new topics and topicref moves on topicset and whether the aggregated authoring solution transforms will</p>

	be able to incorporate sufficient logic. Use cases need to be identified.
or topicsetref	Not replicated in the aggregated document There is some question as to the effect of new topics, and topicref moves on topicsetref and whether the aggregated authoring solution transforms will be able to incorporate sufficient logic. Use cases need to be identified.

Draft for Internal Subcommittee Review