Review K: Bookmap elements

Table of contents

1 Topic and map document types

The Technical Content package contains various document types: concept, glossary entry, glossary group, reference, general task, strict task, and troubleshooting. The package also includes the bookmap document type.

1.1 Bookmap

> **Comment by Eric Sirois**
> Should we add mapresources to the example? As well as updated content model and associated topic.
>
> Robert Anderson, Sept 26 2023
>
> If we did that - we need to be clear that this is an optional element (it comes from a domain and will not be present in every bookmap)

> **Comment by Comtech Services, Inc.**
> where is mapresources discussed? in the maps section?

The DITA bookmap specialization represents the key markup requirements for managing DITA content through book-oriented publication processes, including book metadata and book structures for organizing content.

Purpose

Book maps

> **Comment by Robert D Anderson on Sept 26 2023**
> We need to ensure we consistently use "bookmap" vs "book map". Based on the book map metadata review, we decided to go with "book map" in natural language, and reserve "bookmap" for when we are talking specifically about the specialization module.

enable authors to produce documents that are structured like traditional print-oriented media. They also provide metadata for recording information about the book, including authors, owners, versions, and production history.

Content model

A book map can contain the following document structures:

- Titles
- Metadata
- Front matter (including notices and a preface)

> **Comment by Comtech Services, Inc.**
> I don't see the need for the parenthetical remark. why doesn't back matter say including index and glossary? it seems unnecessary to point out what might be part of front matter.

- Parts
- Chapters
- Appendixes

- Back matter
- Relationship tables

Other components of a book map enable authors to specify that artifacts such as a table of contents (TOC) or an index should be generated.

> **Comment by Comtech Services, Inc.**
> But these are part of the structures above, so they aren't "other" components.

Example

The following code sample contains the markup for a bookmap:

> **Comment by Comtech Services, Inc.**
> should rephrase to indicate it doesn't show everything possible. It seems at least though that everything that's called out above should have a placeholder – parts, backmatter, reltable are all missing. and if we continue to callout out notices and preface, then notices as well. And certainly having a toc would be common in any bookmap.

```
<bookmap id="taskbook">
  <booktitle>
    <mainbooktitle>Product tasks</mainbooktitle>
    <booktitlealt>Tasks and what they do</booktitlealt>
  </booktitle>
  <bookmeta>
    <author>John Doe</author>
    <bookrights>
      <copyrfirst>
        <year>2020</year>
      </copyrfirst>
    </bookrights>
  </bookmeta>
  <frontmatter>
    <preface href="task-preface.dita"/>
  </frontmatter>
  <chapter format="ditamap" href="installing.ditamap"/>
  <chapter href="configuring.dita"/>
  <chapter href="maintaining.dita">
    <topicref href="maintain-storage.dita"/>
    <topicref href="maintain-server.dita"/>
    <topicref href="maintain-database.dita"/>
  </chapter>
  <appendix href="task-appendix.dita"/>
</bookmap>
```

> **Comment by Zoë Lawson on 3 Oct 2023**
> My gut feeling is that we may want to add additional content that defines intentions a bit more.
>
> - Do we want some discussion about why there's no 'cover' element? That bookmap is designed so that the cover gets generated by the processing? Maybe state that if you're building a book to be physically published, it's up to the processing to generate blank pages, end papers, etc.
> - Do we want to list the elements that are specialized with the intention of processors using them to generate lists? toc, indexlist, abbrevlist, etc.
> - Do we want to more concretely describe what we consider a 'book' to be and why we provide these elements? preface usually doesn't have a number associated with it, chapters are sequential, appendices restart the numbering and let you change it, parts have their own numbering. I know it's up to the processor to do these things, but as people do more 'online' and lest published formatting, we might want more intentions described (e.g. what we have for colophon).
> - Do we need to state that there's no 'page break' element and that it's up to the processing/ design to have the layout work?

- I agree that I'd like to see a more explicit bookmap example that can help show some of the 'we expect to see notices in front or back matter' information. Yes, it would be long, but book structures are complicated.
- Do we want a stub bookmeta topic in this section that points you to the full bookmeta section?

2 Bookmap elements

Elements in the bookmap section are used to organize DITA content into book form. They include elements for dividing up content, such as chapter and appendix, as well as metadata specific to publishing.

2.1 Bookmap content elements

The bookmap specialization of `<map>` supports standard book production for collections of DITA topics.

> **Comment by Comtech Services, Inc.**
> A few general comments:
>
> 1 - Usage information isn't present often, but when it is, it seems to be information that not only applies to the element in question, but many others. Example: stating the item might be put in either front or back matter. or the information is a better definition than what is in the shortdescription and might be better put there.
>
> 2 – in all but one topic (amendments) the examples use @href; amendments uses @keyref. Which led me to mark every single occurrence of the statement about how the absence of an href might queue the processor to autogenerate. It's not the href, it's the lack of a reference, which might be made in a keyref or conkeyref equally well. I think all of those comments need to be rewritten to not specifically mention href as the queuing factor. In addition, I think that toc, figurelist, tablelist, and indexlist are expected to be blank and be autogenerated. Should that not be a firm expectation rather than a "might"
>
> 3 – I once again find it difficult to predict when I'm going to see an example in the section and when I'm going to be referred to the overarching bookmap example. Because the bookmap example is extensive, I think it's difficult to find the specific one line example in the overarching example and it would be more helpful to include snippets to show the example in each topic.

2.1.1 <abbrevlist>

The `<abbrevlist>` element references a list of abbreviations.

Processing expectations

When the `@href` attribute is not specified, a processor might generate a list of abbreviations at the specified location in the map.

> **Comment by Comtech Services, Inc.**
> I assume this would be based on <abbreviated-form> elements. Is that too prescriptive to include here? Also see note on amendments about keyref – make this "When a reference is not specified..."?

Specialization hierarchy

The `<abbrevlist>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

For this element, the `@href` attribute references a manual listing for the current element. If the `@href` attribute is not specified, processors can choose to generate an appropriate listing for this element.

> **Comment by Comtech Services, Inc.**
> Not entirely accurate – if a reference is not specified (it could be specified by a keyref, for example, not an href.

Example

The following code sample shows how to reference a list of abbreviations as part of a publication's back matter:

```
<backmatter>
  <booklists>
    <abbrevlist href="abbrev.dita"/>
    <indexlist/>
  </booklists>
</backmatter>
```

2.1.2 <amendments>

The `<amendments>` element either references a list of amendments or changes to the book, or it indicates to a processor that a list of changes should be generated at this location in the book.

Processing expectations

When the `@href` attribute is not specified, a processor might generate a list of amendments or updates at the specified location in the map.

> **Comment by Comtech Services, Inc.**
> from <changehistory-list>, or again? too prescriptive?

Specialization hierarchy

The `<amendments>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0).

Example

The following code sample specifies that a change history list is generated in the publication front matter. The content of the change history list is contained in the DITA resource referenced by `@keyref`.

> **Comment by Comtech Services, Inc.**
> is it confusing that the processing instructions talks about an href attribute and if not present generating automatically; there's technically no "href" here. Perhaps we need to rephrase the processing instructions to be more like the note below – if it doesn't reference a resource. And while this doesn't negate the need to change the notes throughout about the absence of an href, why is this the only example that chose to use a keyref instead?

```
<frontmatter>
  <booklists>
```

```
    <amendments keyref="change-history"/>
  </booklists>
</frontmatter>
```

If the `<amendments>` element did not reference a resource, the change history list would be auto-generated by the processor.

> **Comment by Comtech Services, Inc.**
> In all other topics with a similar comment, this note is at the end of the attributes section.

2.1.3 <appendices>

The `<appendices>` element is an optional container for `<appendix>` elements.

Specialization hierarchy

The `<appendices>` element is specialized from `<topicref>`. It is defined in the bookmap module.

> **Comment by Zoë Lawson on 3 Oct 2023**
> This is more a general comment that I'm sticking here because I thought of it here, but it might go back to <topicref>. This is an optional container. It may or may not contain an href/keyref. Is there a place where we discuss 'containers' or maybe whether or not topicref requires a reference? If this is a container, why isn't it based off of <topichead>?
>
> I'm probably wandering into the user guide realm again, but I'm not sure why I'd use <appendicies>. With the example not including a reference to a topic, it made me wonder how that works in general. Reading the base specification, topicref kind of implies that @href is required...but it isn't. So do we talk about the differences between topicref vs topichead vs topicgroup? (Trying to find it in the main spec I realize it's still in the TBD).
>
> A long ramble to state I wonder if it's worth highlighting that a reference is optional, and maybe provide some sort of link to a summary topic on different types of 'references' in maps.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

Example

The following code sample shows how the `<appendices>` element functions as a container to hold several appendix topics for an HTML publication:

```
<appendices toc="yes" deliveryTarget="html">
  <topicmeta>
    <navtitle>Appendices</navtitle>
  </topicmeta>
  <appendix href="return-codes.dita"/>
  <appendix href="messages.dita"/>
  <appendix href="extra-info.dita"/>
</appendices>
```

2.1.4 <appendix>

An `<appendix>` element references a topic or map as an appendix within a book.

Specialization hierarchy

The `<appendix>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

Example

The following code sample shows how the `<appendix>` element is used to create two appendices for a publication:

> **Comment by Comtech Services, Inc.**
> perhaps being too nitpicky here. but the appendix element doesn't create two appendixes. there are two appendix elements, each creating an appendix.

```
<backmatter>
  <appendix href="intro.dita">
    <topicref href="caring.dita"/>
    <topicref href="feeding.dita"/>
  </appendix>
  <appendix href="setup.dita">
    <topicref href="prereq.dita"/>
    <topicref href="download.dita"/>
  </appendix>
</backmatter>
```

2.1.5 <backmatter>

The `<backmatter>` element is a container for material that follows the main body of a document and any appendices. Back matter might include items such as a colophon, legal notices, and book lists such as a glossary or an index.

> **Comment by Comtech Services, Inc.**
> in frontmatter, the second sentence is put in "Usage Information"

Specialization hierarchy

The `<backmatter>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), universal attributes (39), `@format` (0   ), `@keyref` (0   ), `@scope` (0   ), and `@type` (0   ).

Example

See the example in bookmap (13).

2.1.6 <bibliolist>

The `<bibliolist>` element references a topic that contains a list of bibliographic entries for the publication.

Processing expectations

When the `@href` attribute is not specified, a processor might generate a list of bibliographic entries at the specified location in the map.

> **Comment by Comtech Services, Inc.**
> same comment about href as on amendments and abbrevlist. from what information would this list be generated? cite?

Specialization hierarchy

The `<bibliolist>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0).

For this element, the `@href` attribute references a manual listing for the current element. If the `@href` attribute is not specified, processors can choose to generate an appropriate listing for this element.

> **Comment by Comtech Services, Inc.**
> Same comment about the absence of href isn't the only possible situation for generating a list.

Example

The following code sample shows how a `<bibliolist>` element is used to add a bibliography topic to the publication's back matter:

```
<bookmap>
  <!-- ... -->
  <backmatter>
    <amendments href="updatesToTheBook.dita"/>
    <booklists>
      <trademarklist href="listoftrademarks.dita"/>
      <bibliolist href="bibliography.dita"/>
      <indexlist/>
    </booklists>
  </backmatter>
</bookmap>
```

2.1.7 <bookabstract>

The `<bookabstract>` element references a topic that includes a brief summary of the book content.

Usage information

A book abstract is generally rendered as part of the book's front matter. It is used to help the reader quickly evaluate the book's purpose.

> **Comment by Comtech Services, Inc.**

> is this usage information helpful and necessary? seems like the second sentence could be folded into the short description, and not sure we need to say where it's generally rendered at all.

Specialization hierarchy

The `<bookabstract>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

Example

See the example in bookmap (13).

> **Comment by Comtech Services, Inc.**
> I know that we are doing this generally, but the bookmap example contains so much and if you don't really know where to look for th element, it's hard to find. what's the harm in showing just the frontmatter here with bookabstract highlighted?

2.1.8 <booklibrary>

The `<booklibrary>` element contains information about the library, series, or collection of documents to which a book belongs.

Specialization hierarchy

The `<booklibrary>` element is specialized from `<ph>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: universal attributes (39) and `@keyref` (0   ).

Example

See the example in bookmap (13).

> **Comment by Comtech Services, Inc.**
> maybe a more realistic example here would be helpful. "Books about stuff" doesn't seem worth referring to people to another topic for. Something like:
>
> <booktitle>
>
> <booklibrary>Construction Equipment</booklibrary>
>
> <mainbooktitle>Bulldozer Service Manual</mainbooktitle>
>
> <booktitlealt>Models 1234-5678</booktitlealt>
>
> </booktitle>

2.1.9 <booklist>

A `<booklist>` indicates a list of items within the book.

> **Comment by Comtech Services, Inc.**
> change indicates to references to be parallel to other descriptions? Seems like the first sentence of
> usage information would be more illustrative here.

Usage information

The `<booklist>` element is a general purpose element, designed for use in specializations, that references a topic or map containing a list of items within the book. For example, it could be used to reference a topic that contains a list of authors for the book. When a more specific element is already available, such as `<tablelist>` for a list of tables, use that element instead.

Specialization hierarchy

The `<booklist>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

For this element, the `@href` attribute references a manual listing for the current element. If the `@href` attribute is not specified, processors can choose to generate an appropriate listing for this element.

> **Comment by Comtech Services, Inc.**
> same comment as earlier on href. Why isn't there a processing expectation section that highlights this
> as in other topics?

Example

The following code sample shows how to use `<booklist>` to reference a topic that contains a list of authors of topics in this document:

```
<booklists>
  <toc/>
  <tablelist/>
  <booklist href="authors.dita">
    <topicmeta>
      <navtitle>List of authors</navtitle>
    </topicmeta>
  </booklist>
</booklists>
```

2.1.10 <booklists>

The `<booklists>` element is a container for lists of various kinds within the book.

Processing expectations

The `<booklists>` element indicates to processors that lists are to be rendered or generated at that location in the front or back matter.

Specialization hierarchy

The `<booklists>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), universal attributes (39), `@format` (0   ), `@keyref` (0   ), `@scope` (0   ), and `@type` (0   ).

Example

The following code sample indicates that lists are generated in the front and back matter of the publication.

```
<bookmap>
  <booktitle>
    <mainbooktitle>Sample publication</mainbooktitle>
  </booktitle>
  <frontmatter>
    <booklists>
      <toc>
      <amendments/>
```

> **Comment by**
> **Comtech Services, Inc.**
> I generally see amendments before toc or after table list, but not separating them.

```
      <figurelist/>
      <tablelist/>
    </booklists>
  </frontmatter>
  ...
  <backmatter>
    <booklists>
      <abbrevlist/>
      <glossarylist/>
      <indexlist/>
    </booklists>
  </backmatter>
</bookmap>
```

2.1.11 <bookmap>

The `<bookmap>` element is a map specialization that is used to configure DITA topics as a traditional book.

> **Comment by Comtech Services, Inc.**
> recommend "assemble" rather than "configure"

Usage information

Book maps consist of references to topics organized as book content. The topic references therefore are labeled according to the book components they point to, such as book title, front matter, chapter, and appendix.

Specialization hierarchy

The `<bookmap>` element is specialized from `<map>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: architectural attributes (44), common map attributes (44), universal attributes (39), `@format` (0   ), `@scope` (0   ), and `@type` (0   ).

Example

The following code sample shows how a `<bookmap>` can be used to organize content into a common book structure:

> **Comment by Comtech Services, Inc.**
> I understand the example is generic. Could we just put Book Library and Book Title as the values in
> <booklibrary> and <mainbooktitle>.

```
<bookmap xml:lang="en-us">
  <booktitle>
    <booklibrary>Books about stuff</booklibrary>
    <mainbooktitle>A book about one thing</mainbooktitle>
  </booktitle>
  <bookmeta>
    <bookrights>
      <copyrfirst><year>2019</year></copyrfirst>
      <copyrlast><year>2023</year></copyrlast>
      <bookowner>
        <organization>OASIS</organization>
      </bookowner>
    </bookrights>
  </bookmeta>
  <frontmatter>
    <booklists>
      <toc/>
      <figurelist/>
      <tablelist/>
    </booklists>
    <bookabstract href="MyBookAbstract.dita"/>
    <preface href="preface.dita"></preface>
  </frontmatter>
  <chapter href="chapter1.dita">
    <topicref href="subchap1.dita"></topicref>
  </chapter>
  <chapter href="chapter2.dita">
    <topicref href="subchap2.dita"></topicref>
  </chapter>
  <appendix href="app1.dita">
    <topicref href="insideApp1.dita"></topicref>
  </appendix>
  <appendix href="app2.dita">
    <topicref href="insideApp2.dita"></topicref>
  </appendix>
  <backmatter>
    <amendments href="updatesToTheBook.dita"/>
```

> **Comment by
> Comtech Services, Inc.**
> Recommend making the file "RevisionHistorytTable.dita" as that's what most of my clients
> use amendments for.

```
    <booklists>
      <glossarylist href="listofterms.dita"/>
      <trademarklist href="listoftrademarks.dita"/>
      <indexlist/>
    </booklists>
  </backmatter>
</bookmap>
```

2.1.12 <booktitle>

The `<booktitle>` element contains the title information for a book, including the library title, main title, subtitle, and any other titles.

> **Comment by Comtech Services, Inc.**

> "...main title, and any other subtitles." Since all other titles are put into booktitlealt, it seems misleading of sorts to have four categories here and only three elements.

Specialization hierarchy

The `<booktitle>` element is specialized from `<title>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: ID and conref attributes (40), localization attributes (40), `@base` (0), `@class` (0), `@outputclass` (0), and `@rev` (0).

Example

See the example in bookmap (13).

2.1.13 <booktitlealt>

The `<booktitlealt>` element contains an alternative title, subtitle, or short title for a book.

Specialization hierarchy

The `<booktitlealt>` element is specialized from `<ph>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: universal attributes (39) and `@keyref` (0).

Example

The following code sample shows a `<booktitlealt>` element is used to provide a shorter alternative to the main book title:

> **Comment by Comtech Services, Inc.**
> Recommend a less informal mainbooktitle. Maybe "Product Z Installation, Operation, and Maintenance Manual". Also, while a valid use case, I've never had anyone use booktitlealt this way, instead always using it as a subtitle.

```
<bookmap>
  <booktitle>
    <mainbooktitle>This is my big and fancy book about Product Z</mainbooktitle>
    <booktitlealt>Using Product Z</booktitlealt>
  </booktitle>
  <!-- ... -->
</bookmap>
```

2.1.14 <chapter>

The `<chapter>` element references a topic or map as a chapter within a book.

Specialization hierarchy

The `<chapter>` element is specialized from `<topicref>`. It is defined in the bookmap module.

> **Comment by Comtech Services, Inc.**

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

Example

The following code sample shows how the `<chapter>` element is used to create two chapters within a publication:

```
<bookmap>
  <booktitle>
    <mainbooktitle>A book about one thing</mainbooktitle>
  </booktitle>
  <!-- ... metadata and front matter ... -->
  <chapter href="intro.dita">
    <topicref href="caring.dita"/>
    <topicref href="feeding.dita"/>
  </chapter>
  <chapter href="setup.dita">
    <topicref href="prereq.dita"/>
    <topicref href="download.dita"/>
  </chapter>
  <!-- ... more chapters and back matter ... -->
</bookmap>
```

2.1.15 <colophon>

The `<colophon>` element references a topic that describes how the document was created.

Usage information

In publishing, a colophon describes details of the production of a book. This information generally includes the typefaces used, and often the names of their designers; the paper, ink and details of the binding materials and methods might also receive mention. In the case of technical books, a colophon might specify the software used to prepare the text and diagrams for publication. The `<colophon>` can appear in the `<frontmatter>` or `<backmatter>`.

Specialization hierarchy

The `<colophon>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

Example

The following code sample shows how the `<colophon>` element is used to add details about the book's production into the back matter:

```
<bookmap>
  <title>Sample book</title>
  <!-- ... -->
  <backmatter>
    <!-- ... other back matter ... -->
    <colophon href="ProductionNotes.dita" />
  </backmatter>
</bookmap>
```

2.1.16 <dedication>

The `<dedication>` element references a topic that contains a dedication for the book, such as to a person or group.

Specialization hierarchy

The `<dedication>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

Example

The following code sample shows how a `<dedication>` element is used to supply content for the dedication within the book's frontmatter.

```
<frontmatter>
  <dedication href="dtm.dita">
    <topicmeta>
      <navtitle>Dedicated to Mother</navtitle>
    </topicmeta>
  </dedication>
</frontmatter>
```

2.1.17 <draftintro>

The `<draftintro>` element references a topic used as an introduction to the current draft of a book.

> **Comment by Zoë Lawson on 3 Oct 2023**
> This may be silly, but in my brain, draft = not published. So if this is just for a draft, potentially automatically excluded from output if draft = no. However, I think this might be possibly also used for release or edition notices?

Specialization hierarchy

The `<draftintro>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

Example

The following code sample uses `<draftintro>` to provide an introductory draft section within the frontmatter of the book.

```
<frontmatter>
  <draftintro href="introducing.dita">
    <topicmeta>
      <navtitle>Introduction to this draft</navtitle>
    </topicmeta>
  </draftintro>
  <!-- ... -->
</frontmatter>
```

2.1.18 <figurelist>

The `<figurelist>` element either references a topic that contains a list of figures in the book, or it indicates to a processor that a list of figures should be generated at this location in the book.

Processing expectations

When the `@href` attribute is not specified, a processor might generate a list of figures at the specified location in the map.

> **Comment by Comtech Services, Inc.**
> Isn't this an expectation that the processor would generate a list? indicate that it would be based on <fig> elements?

> **Comment by Comtech Services, Inc.**
> rewrite for all cases that might reference a list of figures.

Specialization hierarchy

The `<figurelist>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

For this element, the `@href` attribute references a manual listing for the current element. If the `@href` attribute is not specified, processors can choose to generate an appropriate listing for this element.

> **Comment by Comtech Services, Inc.**
> same comment as previous about href. also, should be this be stronger than "can choose"

Example

See the example in bookmap (13).

2.1.19 <frontmatter>

The `<frontmatter>` element is a container for the material that precedes the main body of a document.

Usage information

The front matter might include items such as an abstract, notices, a preface, and various types of book lists such as a `<toc>`, `<tablelist>`, or `<figurelist>`.

> **Comment by Comtech Services, Inc.**
> In backmatter, this is part of the short description.

Specialization hierarchy

The `<frontmatter>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), universal attributes (39), `@format` (0   ), `@keyref` (0   ), `@scope` (0   ), and `@type` (0   ).

Example

See the example in bookmap (13).

2.1.20 <glossarylist>

The `<glossarylist>` element either references a topic that contains a list of glossary entries in the book, or it indicates to a processor that a list of glossary entries should be generated at this location in the book.

Processing expectations

When the `@href` attribute is not specified, a processor might generate a list of glossary entries at the specified location in the map.

> **Comment by Comtech Services, Inc.**
> same comment about href; include that this could be generated from <term> or <abbreviated-form> elements?

Specialization hierarchy

The `<glossarylist>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

For this element, the `@href` attribute references a manual listing for the current element. If the `@href` attribute is not specified, processors can choose to generate an appropriate listing for this element.

> **Comment by Comtech Services, Inc.**

same comment about href

Example

See the example in bookmap (13).

2.1.21 <indexlist>

The <indexlist> element either references a topic that contains an index for the book, or it indicates to a processor that an index should be generated at this location in the book.

Processing expectations

When the @href attribute is not specified, a processor might generate an index at the specified location in the map. If the @href attribute is specified, the referenced topic or map contains a manually created index.

Comment by Comtech Services, Inc.
same comment about href, but also who would actually create a manual index? doesn't this negate a large part of the reuse benefits of dita? different output formats might result in different page divisions and a manual index would be wrong in one case then. Also, isn't this an expectation, not just a might?

Specialization hierarchy

The <indexlist> element is specialized from <topicref>. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and @keyref (0).

For this element, the @href attribute references a manual listing for the current element. If the @href attribute is not specified, processors can choose to generate an appropriate listing for this element.

Comment by Comtech Services, Inc.
same comment about href. In this case is it a "can" or more of an expectaton?

Example

See the example in bookmap (13).

2.1.22 <mainbooktitle>

The <mainbooktitle> element contains the primary title for a book.

Specialization hierarchy

The <mainbooktitle> element is specialized from <ph>. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: universal attributes (39) and @keyref (0).

Example

See the example in bookmap (13).

2.1.23 <notices>

The `<notices>` element references a topic that contains special notice information, such as legal notices about supplementary copyrights and trademarks associated with the book.

Usage Information

Use a `<notices>` element within either the front matter or back matter of a book.

> **Comment by Comtech Services, Inc.**
> this type of usage information doesn't appear on all the elements for which this is true? Should it?

Specialization hierarchy

The `<notices>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0).

Example

The following code sample shows how to use a notices topic to include legal content:

```
<backmatter>
  <notices href="notices.dita" navtitle="Legal notices">
    <topicmeta>
      <navtitle>Legal notices</navtitle>
    </topicmeta>
  </notices>
  <booklists>
    <!-- Index, glossary, or other lists -->
  </booklists>

                          Comment by
                          Comtech Services, Inc.
  not sure why we need th context of booklists in this example?

</backmatter>
```

2.1.24 <part>

The `<part>` element references a topic or a map that acts as a part within a book.

Usage information

Use `<part>` to divide a document's chapters into logical groupings. For example, in a document that contains both guide and reference information, you can define two parts, one containing the guide information and the other containing the reference information.

Specialization hierarchy

The `<part>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

Example

The following code sample shows how `<part>` elements are used to group chapters in order to divide a book into two major sections for task and reference material:

```
<bookmap>
  <title>Using and maintaining Product Zed</title>
  <!-- ... metadata and front matter ... ->
  <part href="taskguide.dita">
    <chapter href="intro.dita">
      <topicref href="caring.dita"/>
      <topicref href="feeding.dita"/>
    </chapter>
    <chapter href="setup.dita">
      <topicref href="prereq.dita"/>
      <topicref href="download.dita"/>
    </chapter>
  </part>
  <part href="reference.dita">
    <chapter href="commands.dita">
      <topicref href="care.dita"/>
      <topicref href="feed.dita"/>
    </chapter>
    <chapter href="apis.dita">
      <topicref href="acare.dita"/>
      <topicref href="afeed.dita"/>
    </chapter>
  </part>
</bookmap>
```

2.1.25 <preface>

The `<preface>` element references a topic or map that contains introductory information about a book, such as the purpose and structure of the document.

Specialization hierarchy

The `<preface>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

Example

See the example in bookmap (13).

2.1.26 <tablelist>

The `<tablelist>` element either references a topic that contains a list of tables in the book, or it indicates to a processor that a list of tables should be generated at this location in the book.

Processing expectations

When the `@href` attribute is not specified, a processor might generate a list of tables at the specified location in the map.

> **Comment by Comtech Services, Inc.**
> same comment about href. also, isn't this an expectation, not a might?

Specialization hierarchy

The `<tablelist>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

For this element, the `@href` attribute references a manual listing for the current element. If the `@href` attribute is not specified, processors can choose to generate an appropriate listing for this element.

> **Comment by Comtech Services, Inc.**
> same comment about href and whether this should be stronger than "can choose"

Example

See the example in bookmap (13).

2.1.27 <toc>

The `<toc>` element either references a topic that contains a table of contents for the book, or it indicates to a processor that a table of contents should be generated at this location in the book.

Processing expectations

When the `@href` attribute is not specified, a processor might generate a table of contents at the specified location in the map. If the `@href` attribute is specified, the referenced topic or map contains a manually created table of contents.

> **Comment by Comtech Services, Inc.**
> same comments about href and that I think this is an expectation, not a might.

Specialization hierarchy

The `<toc>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

For this element, the `@href` attribute references a manual listing for the current element. If the `@href` attribute is not specified, processors can choose to generate an appropriate listing for this element.

> **Comment by Comtech Services, Inc.**
> same comments as before about href and "can choose"

Example

See the example in bookmap (13).

2.1.28 <trademarklist>

The `<trademarklist>` element either references a topic that contains a list of trademarks in the book, or it indicates to a processor that a list of trademarks should be generated at this location in the book.

Processing expectations

When the `@href` attribute is not specified, a processor might generate a list of trademarks at the specified location in the map.

> **Comment by Comtech Services, Inc.**
> same comments on href; specify the autogenerated list would be based on tm?

Specialization hierarchy

The `<trademarklist>` element is specialized from `<topicref>`. It is defined in the bookmap module.

Attributes

The following attributes are available on this element: common map attributes (44), link-relationship attributes (45), universal attributes (39), and `@keyref` (0   ).

For this element, the `@href` attribute references a manual listing for the current element. If the `@href` attribute is not specified, processors can choose to generate an appropriate listing for this element.

> **Comment by Comtech Services, Inc.**
> same comment on href

Example

See the example in bookmap (13).

Appendix A Aggregated RFC-2119 statements

This appendix contains all the normative statements from the DITA for Technical Content 2.0 specification. They are aggregated here for convenience in this non-normative appendix.

Appendix B Attributes

This section contains definitions for commonly-used attributes. If an attribute is defined differently on a specific element, that information is covered in the topic for the specific element.

> **Comment by Kristen J Eberlein on 29 December 2021**
>
> Add a brief overview of the fact that some specific attributes are overloaded – and have different meanings depending on what element they are specified upon.

Appendix B.1 Attribute groups

Many of the attributes used on DITA elements are defined in attribute groups. These attribute groups are used both in the grammar files and the specification,

Architectural attributes

This group contains a set of attributes that are defined for document-level elements such as `<topic>` and `<map>`.

@DITAArchVersion (architectural attributes)
Specifies the version of the DITA architecture that is in use. This attribute is in the namespace `http://dita.oasis-open.org/architecture/2005/`. This attribute is specified in the topic and map modules, and it uses a default value of the current version of DITA. The current default is "2.0".

@specializations (architectural attributes)
Specifies the attribute-domain specializations that are included in the document-type shell. This attribute is set as a default within the document-type shell. The value varies depending on what domains are integrated into the document-type shell. For example, a grammar file that includes the specialized attributes `@audience`, `@deliveryTarget`, and `@newBaseAtt` would set the value to `@props/audience @props/deliveryTarget @base/newBaseAtt`.

@xmlns:ditaarch (architectural attributes)
Declares the default DITA namespace. This namespace is declared as such in the RNG modules for `<topic>` and `<map>`, but it is specified as an attribute in the equivalent DTD-based modules. The value is fixed to "http://dita.oasis-open.org/architecture/2005/".

Common map attributes

This group contains attributes that are frequently used on map elements.

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> I've added draft comments to the attribute definitions in this section that explain how the attribute is defined in the "DITA map attributes" topic.

@cascade (common map attributes)

Specifies how metadata attributes cascade within a map. The specification defines the following values:

merge
> Indicates that the metadata attributes cascade, and that the values of the metadata attributes are additive. This is the processing default for the `@cascade` attribute.

nomerge
> Indicates that the metadata attributes cascade, but that they are not additive for `<topicref>` elements that specify a different value for a specific metadata attribute. If the cascading value for an attribute is already merged based on multiple ancestor elements, that merged value continues to cascade until a new value is encountered. That is, setting `cascade="nomerge"` does not undo merging that took place on ancestor elements.

Processors can also define custom, implementation-specific tokens for this attribute.

See Cascading of metadata attributes in a DITA map for more information about how this attribute interacts with metadata attributes.

@chunk (common map attributes)
Specifies how a processor should render a map or branch of a map. For example, it can be used to specify that individual topic documents should be rendered as a single document, or that a single document with multiple topics should be rendered as multiple documents.

The following values are valid:

combine
> Instructs a processor to combine the referenced source documents for rendering purposes. This is intended for cases where a publishing process normally results in a single output artifact for each source XML document.

split
> Instructs a processor to split each topic from the referenced source document into its own document for rendering purposes. This is intended for cases where a publishing process normally results in a single output artifact for each source XML document, regardless of how many DITA topics exist within each source document.

Processors can also define custom, implementation-specific tokens for this attribute.

For a detailed description of the `@chunk` attribute and its usage, see Chunking.

@collection-type (common map attributes)
Specifies how topics or links relate to each other. The processing default is "unordered", although no default is specified in the OASIS-provided grammar files. The following values are valid:

unordered
> Indicates that the order of the child topics is not significant.

sequence
> Indicates that the order of the child topics is significant. Output processors will typically link between them in order.

choice
> Indicates that one of the children should be selected.

family
> Indicates a tight grouping in which each of the referenced topics not only relates to the current topic but also relate to each other.

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> Here is the content from the "DITA map attributes" topic:

@collection-type

The `@collection-type` attribute specifies how the children of a `<topicref>` element relate to their parent and to each other. This attribute, which is set on the parent element, typically is used by processors to determine how to generate navigation links in the rendered topics. For example, a `@collection-type` value of "sequence" indicates that children of the specifying `<topicref>` element represent an ordered sequence of topics; processors might add numbers to the list of child topics or generate next/previous links for online presentation. This attribute is available in topics on the `<linklist>` and `<linkpool>` elements, where it has the same behavior. Where the `@collection-type` attribute is available on elements that cannot directly contain elements, the behavior of the attribute is undefined.

Comment by Kristen J Eberlein on 28 September 2022

In the definitions of the supported values, do we want to refer to "resources" instead of "topics"? Since we specify that `@collection-type` specifies "how topics **or links** relate to each other" ...

@keyscope (common map attributes)

Specifies that the element marks the boundaries of a key scope.

See [STUB CONTENT](#) (56) for information on using this attribute.

Comment by Kristen J Eberlein on 28 September 2022

Here is the content from the "DITA map attributes" topic:

@keyscope

Defines a new scope for key definition and resolution, and gives the scope one or more names. For more information about key scopes, see [Indirect key-based addressing](#).

@linking (common map attributes)

Specifies linking characteristics of a topic specific to the location of this reference in a map. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see [Cascading of metadata attributes in a DITA map](#)).

Comment by robander on Dec 28 2021

The text below matches [1.3 spec text](#) but I'm nervous about "cannot link" type definition. It's describing how to generate links based on the current context in the map - it's not describing what the topic itself is allowed to link to, which is how I interpret "can".

The following values are valid:

targetonly

A topic can only be linked to and cannot link to other topics.

sourceonly

A topic cannot be linked to but can link to other topics.

normal

A topic can be linked to and can link to other topics. Use this to override the linking value of a parent topic.

none

A topic cannot be linked to or link to other topics.

-dita-use-conref-target

    See [Using the -dita-use-conref-target value](#) for more information.

<div style="border:1px solid #000; background:#d6eef2; padding:10px;">

Comment by Kristen J Eberlein on 28 September 2022

Here is the content from the "DITA map attributes" topic:

@linking

    By default, the relationships between the topics that are referenced in a map are reciprocal:

- Child topics link to parent topics and vice versa.
- Next and previous topics in a sequence link to each other.
- Topics in a family link to their sibling topics.
- Topics referenced in the table cells of the same row in a relationship table link to each other. A topic referenced within a table cell does not (by default) link to other topics referenced in the same table cell.

This behavior can be modified by using the `@linking` attribute, which enables an author or information architect to specify how a topic participates in a relationship. The following values are valid:

linking="none"
    Specifies that the topic does not exist in the map for the purposes of calculating links.

linking="sourceonly"
    Specifies that the topic will link to its related topics but not vice versa.

linking="targetonly"
    Specifies that the related topics will link to it but not vice versa.

linking="normal"
    Default value. It specifies that linking will be reciprocal (the topic will link to related topics, and they will link back to it).

Authors also can create links directly in a topic by using the `<xref>` or `<link>` elements, but in most cases map-based linking is preferable, because links in topics create dependencies between topics that can hinder reuse.

Note that while the relationships between the topics that are referenced in a map are reciprocal, the relationships merely *imply* reciprocal links in generated output that includes links. The rendered navigation links are a function of the presentation style that is determined by the processor.

</div>

@processing-role (common map attributes)

    Specifies whether the referenced resource is processed normally or treated as a resource that is only included in order to resolve references, such as key or content references. The following values are valid:

normal
    Indicates that the resource is a readable part of the information set. It is included in navigation and search results. This is the default value for the `<topicref>` element.

resource-only
    Indicates that the resource should be used only for processing purposes. It is not included in navigation or search results, nor is it rendered as a topic. This is the default value for the `<keydef>` element.

-dita-use-conref-target
>   See Using the -dita-use-conref-target value for more information.

If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@search (common map attributes)
>   Specifies whether the target is available for searching. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see Cascading of metadata attributes in a DITA map). The following values are valid: "yes", "no", and "-dita-use-conref-target".

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> Here is the content from the "DITA map attributes" topic:
>
> **@search**
> >   Specifies whether the topic is included in search indexes.

@subjectrefs (common map attributes)
>   Specifies one or more keys that are each defined by a subject definition in a subject scheme map. Multiple values are separated by white space.

@toc (common map attributes)
>   Specifies whether a topic appears in the table of contents (TOC) based on the current map context. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see Cascading of metadata attributes in a DITA map). The following values are valid:

> **yes**
> >   The topic appears in a generated TOC.

> **no**
> >   The topic does not appear in a generated TOC.

> **-dita-use-conref-target**
> >   See STUB CONTENT (56) for more information.

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> Here is the content from the "DITA map attributes" topic:
>
> **@toc**
> >   Specifies whether topics are excluded from navigation output, such as a Web site map or an online table of contents. By default, `<topicref>` hierarchies are included in navigation output; relationship tables are excluded.

Complex table attributes

This group includes attributes that are defined on complex table elements. Unless other noted, these attributes are part of the OASIS Exchange Table Model. Complex table elements typically use only a subset of the attributes that are defined in this group.

@align (complex table attributes)
>   Specifies the horizontal alignment of text in table entries. The following values are valid:

> **left**
> >   Indicates left alignment of the text.

right
> Indicates right alignment of the text.

center
> Indicates center alignment of the text.

justify
> Justifies the contents to both the left and the right.

char
> Indicates character alignment. The text is aligned with the first occurrence of the character specified by the `@char` attribute.

-dita-use-conref-target
> See Using the -dita-use-conref-target value for more information.

The `@align` attribute is available on the following table elements: `<colspec>`, `<entry>`, and `<tgroup>`.

@char (complex table attributes)
Specifies the alignment character, which is the character that is used for aligning the text in table entries. This attribute applies when `align="char"`. A value of "" (the null string) means there is no aligning character.

For example, if `align="char"` and `char="."` are specified, then text in the table entry aligns with the first occurrence of the period within the entry. This might be useful if decimal alignment is required.

The `@char` attribute is available on the following table elements: `<colspec>` and `<entry>`.

@charoff (complex table attributes)
Specifies the horizontal offset of the alignment character that is specified by the `@char` attribute. The value is a greater-than-zero number that is less than or equal to 100. It represents the percentage of the current column width by which the text is offset to the left of the alignment character.

For example, if `align="char"`, `char="."`, and `charoff="50"` are all specified, then text in the table entry is aligned 50% of the distance to the left of the first occurrence of the period character within the table entry.

The `@charoff` attribute is available on the following table elements: `<colspec>` and `<entry>`.

@colsep (complex table attributes)
Specifies whether to render column separators between table entries. The following values are valid: "0" (no separators) and "1" (separators).

The `@colsep` attribute is available on the following table elements: `<colspec>`, `<entry>`, `<table>`, and `<tgroup>`.

@rowheader (complex table attributes)
Specifies whether the entries in the respective column are row headers. The following values are valid:

firstcol
> Indicates that entries in the first column of the table are row headers. This applies when the `@rowheader` attribute is specified on the `<table>` element.

headers
> Indicates that entries of the column that is described using the `<colspec>` element are row headers. This applies when the `@rowheader` attribute is specified on the `<colspec>` element.

norowheader
> Indicates that entries in the first column are not row headers. This applies when the `@rowheader` attribute is specified on the `<table>` element.

-dita-use-conref-target
> See Using the -dita-use-conref-target value for more information.

> **Note** This attribute is not part of the OASIS Exchange Table Model upon which DITA tables are based. Some processors or output formats might not support all values.

The `@rowheader` attribute is available on the following table elements: `<table>` and `<colspec>`.

@rowsep (complex table attributes)
Specifies whether to render row separators between table entries. The following values are valid: "0" (no separators) and "1" (separators).

The `@rowsep` attribute is available on the following table elements: `<colspec>`, `<entry>`, `<row>`, `<table>`, and `<tgroup>`.

@valign (complex table attributes)
Specifies the vertical alignment of text in table entries. The following values are valid:

bottom
> Indicates that text is aligned with the bottom of the table entry.

middle
> Indicates that text is aligned with the middle of the table entry.

top
> Indicates that text is aligned with the top of the table entry.

-dita-use-conref-target
> See Using the -dita-use-conref-target value for more information.

The `@valign` attribute is available on the following table elements: `<entry>`, `<tbody>`, `<thead>`, and `<row>`.

Data-element attributes

This group contains attributes that are defined on the `<data>` element and its specializations.

@datatype (data-element attributes)
Specifies the type of data contained in the `@value` attribute or within the `<data>` element. A typical use of `@datatype` will be the identifying URI for an XML Schema datatype.

@name (data-element attributes)
Defines a unique name for the object.

> **Comment by robander**
> Do we need to specify the scope of "unique" here?

@value (data-element attributes)
Specifies a value associated with the current property or element.

Date attributes

This group contains attributes that take date values. They are defined on metadata elements that work with date information:

@expiry (date attributes)
> Specifies the date when the information should be retired or refreshed. The date is specified using the ISO 8601 format: *YYYY-MM-DD*, where *YYYY* is the year, *MM* is the month (01 to 12), and *DD* is the day (01-31).

@golive (date attributes)
> Specifies the publication or general availability (GA) date. The date is specified using the ISO 8601 format: *YYYY-MM-DD*, where *YYYY* is the year, *MM* is the month (01 to 12), and *DD* is the day (01-31).

Display attributes

This group contains attributes that affect the rendering of many elements.

@expanse (display attributes)
> Specifies the horizontal placement of the element. The following values are valid:

> **column**
>> Indicates that the element is aligned with the current column margin.

> **page**
>> Indicates that the element is placed on the left page margin for left-to-right presentation or the right page margin for right-to-left presentation.

> **spread**
>> Indicates that the object is rendered across a multi-page spread. If the output format does not have anything that corresponds to spreads, then "spread" has the same meaning as "page".

> **textline**
>> Indicates that the element is aligned with the left (for left-to-right presentation) or right (for right-to-left presentation) margin of the current text line and takes indentation into account.

> **-dita-use-conref-target**
>> See Using the -dita-use-conref-target value for more information.

> For `<table>`, in place of the `@expanse` attribute that is used by other DITA elements, the `@pgwide` attribute is used in order to conform to the OASIS Exchange Table Model.

> Some processors or output formats might not support all values.

@frame (display attributes)
> Specifies which portion of a border surrounds the element. The following values are valid:

> **all**
>> Indicates that a line is rendered at the top, bottom, left, and right of the containing element.

> **bottom**
>> Indicates that a line is rendered at the bottom of the containing element.

> **none**
>> Indicates that no lines are rendered.

> **sides**
>> Indicates that a line is rendered at the left and right of the containing element.

> **top**
>> Indicates that a line is rendered at the top of the containing element.

> **topbot**
>> Indicates that a line is rendered at the top and bottom of the containing element.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

Some processors or output formats might not support all values.

@scale (display attributes)
Specifies the percentage by which fonts are resized in relation to the normal text size. The value of this attribute is a positive integer. When used on `<table>` or `<simpletable>`, the following values are valid: "50", "60", "70", "80", "90", "100", "110", "120", "140", "160", "180", "200", and -dita-use-conref-target (56).

This attribute is primarily useful for print-oriented display. Some processors might not support all values.

If the `@scale` attribute is specified on an element that contains an image, the image is not scaled. The image is scaled **only** if a scaling property is explicitly specified for the `<image>` element.

ID and conref attributes

This group contains the attributes that enable the naming and referencing of elements.

@conaction
Specifies how the element content will be pushed into a new location. The following values are valid:

mark
The element acts as a marker when pushing content before or after the target, to help ensure that the push action is valid. The element with `conaction="mark"` also specifies the target of the push action with `@conref`. Content inside of the element with `conaction="mark"` is not pushed to the new location.

pushafter
Content from this element is pushed after the location specified by `@conref` on the element with `conaction="mark"`. The element with `conaction="pushafter"` is the first sibling element after the element with `conaction="mark"`.

pushbefore
Content from this element is pushed before the location specified by `@conref` on the element with `conaction="mark"`. The element with `conaction="pushbefore"` is the first sibling element before the element with `conaction="mark"`.

pushreplace
Content from this element replaces any content from the element referenced by the `@conref` attribute. A second element with `conaction="mark"` is not used when using `conaction="pushreplace"`.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

See STUB CONTENT (56) for examples and details about the syntax.

@conkeyref
Specifies a key name or a key name with an element ID that acts as an indirect reference to reusable content. The referenced content is used in place of the content of the current element. See STUB CONTENT (56) for more details about the syntax and behaviors.

@conref
Specifies a URI that references a DITA element. The referenced content is used in place of the content of the current element. See STUB CONTENT (56) for examples and details about the syntax.

@conrefend

Specifies a URI that references the last element in a sequence of elements, with the first element of the sequence specified by `@conref`. The referenced sequence of elements is used in place of the content of the current element. See STUB CONTENT (56) for examples and details about the syntax.

@id

Specifies an identifier for the current element. This ID is the target for references by `@href` and `@conref` attributes and for external applications that refer to DITA or LwDITA content. This attribute is defined with the XML data type NMTOKEN, except where noted for specific elements within the language reference.

See id attribute for more details.

Inclusion attributes

This group includes attributes defined on `<include>` and its specializations:

> **Comment by Kristen J Eberlein on 28 September 2002**
>
> What is specialized from `<include>`? Both base (if any) and technical content ...

@encoding (inclusion attributes)

> **Comment by Kristen J Eberlein on 29 April 2019**
>
> Can we replace "should" in the following definition?

Specifies the character encoding to use when translating the character data from the referenced content. The value should be a valid encoding name. If not specified, processors may make attempts to automatically determine the correct encoding, for example using HTTP headers, through analysis of the binary structure of the referenced data, or the `<?xml?>` processing instruction when including XML as text. The resource should be treated as UTF-8 if no other encoding information can be determined.

When `parse="xml"`, standard XML parsing rules apply for the detection of character encoding. The necessity and uses of `@encoding` for non-standard values of `@parse` are implementation-dependent.

@parse (inclusion attributes)

Specifies the processing expectations for the referenced resource. Processors must support the following values:

text

The contents should be treated as plain text. Reserved XML characters should be displayed, and not interpreted as XML markup.

xml

The contents of the referenced resource should be treated as an XML document, and the referenced element should be inserted at the location of the `<include>` element. If a fragment identifier is included in the address of the content, processors must select the element with the specified ID. If no fragment identifier is included, the root element of the referenced XML document is selected. Any grammar processing should be performed during resolution, such that default attribute values are explicitly populated. Prolog content must be discarded.

It is an error to use `parse="xml"` anywhere other than within `<foreign>` or a specialization thereof.

Processors may support other values for the `@parse` attribute with proprietary processing semantics. Processors should issue warnings and use `<fallback>` when they encounter unsupported `@parse` values. Non-standard `@parse` instructions should be expressed as URIs.

Note Proprietary `@parse` values will likely limit the portability and interoperability of DITA content, so should be used with care.

Link relationship attributes

This group contains attributes whose values can be used for representing navigational relationships.

@format (link-relationship attributes)
Specifies the format of the resource that is referenced. See STUB CONTENT (56) for detailed information on supported values and processing implications.

@href (link-relationship attributes)
Specifies a reference to a resource. See STUB CONTENT (56) for detailed information on supported values and processing implications.

@scope (link-relationship attributes)
Specifies the closeness of the relationship between the current document and the referenced resource. The following values are valid: "local", "peer", "external", and "-dita-use-conref-target".

See STUB CONTENT (56) for detailed information on supported values and processing implications.

@type (link-relationship attributes)
Describes the target of a reference. See STUB CONTENT (56) for detailed information on supported values and processing implications.

Localization attributes

> **Comment by Kristen J Eberlein on 29 September 2022**
>
> The definition of the localizations attribute matches how they are described in the architectural topics. Wherever possible, the definition is reused. Where it is not reused (because the definition in the archSpec topics is in a shortdesc), I've checked to ensure that wording is identical.

This group contains the attributes that are related to translation and localization.

@dir

Identifies or overrides the text directionality. The following values are valid:

lro
Indicates an override of the Unicode Bidirectional Algorithm, forcing the element into left-to-right mode.

ltr
Indicates left-to-right.

rlo
Indicates an override of the Unicode Bidirectional Algorithm, forcing the element into right-to-left mode.

rtl

Indicates right-to-left.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

See The dir attribute for more information.

@translate

Specifies whether the content of the element should be translated. The following values are valid: "yes", "no", and "-dita-use-conref-target".

See Element-by-element recommendations for translators for suggested processing defaults for each element.

> **Comment by Kristen J Eberlein on 31 December 2021**
>
> Does Element-by-element recommendations for translators really provide suggested processing defaults for each element? I thought it covered whether an element was block or in-line and whether there were considerations that translators needed to be aware of.

@xml:lang

Specifies the language and optional locale of the content that is contained in an element. Valid values are language tokens or the null string. The `@xml:lang` attribute and its values are described in the Extensible Markup Language 1.0 specification, fifth edition.

> **Comment by Kristen J Eberlein on 29 September 2022**
>
> Do we also want to direct readers to the architectural topics about the `@xml:lang` attribute?

Metadata attributes

This group contains common metadata attributes: `@base`, `@importance`, `@props`, `@rev`, and `@status`. The `@base` and `@props` attributes can be specialized.

@base

Specifies metadata about the element. It is often used as a base for specialized attributes that have a simple syntax for values, but which are not conditional processing attributes.

The `@base` attribute takes a space-delimited set of values. However, when serving as a container for generalized attributes, the attribute values will be more complex. See Attribute generalization for more details.

@importance

Specifies the importance or priority that is assigned to an element. The following values are valid: "default", "deprecated", "high", "low", "normal", "obsolete", "optional", "recommended", "required", "urgent", and "-dita-use-conref-target". This attribute is not used for conditional processing, although applications might use the value of the `@importance` attribute to highlight elements. For example, in steps of a task topic, the value of the `@importance` attribute indicates whether a step is optional or required.

> **Comment by Kristen J Eberlein on 29 September 2022**
>
> I think the phrase "to highlight elements" is a little off. Maybe "render generated text"? And how about adding "Processors often add text or images to ensure that readers of the generated content understand whether the step is optional or required." to the end of the example?

@props
Specifies metadata about the element. New attributes can be specialized from the `@props` attribute. This attribute supports conditional processing. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

The `@props` attribute takes a space-delimited set of values. However, when serving as a container for generalized attributes, the attribute values will be more complex. See Attribute generalization for more details.

@rev
Specifies a revision level of an element that identifies when the element was added or modified. It can be used to flag outputs when it matches a run-time parameter. It cannot be used for filtering nor is it sufficient to be used for version control. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

> **Comment by Kristen J Eberlein on 29 September 2022**
>
> I want to tweak this. How about the following? Also, neither definition describes what values are permitted.
>
> Specifies metadata that identifies when the element was added or the content of the element was modified. The `@rev` attribute can be used for flagging. It cannot be used for filtering nor is it sufficient to be used for version control. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@status
Specifies the modification status of the element. The following values are valid: "new", "changed", "deleted", "unchanged", and "-dita-use-conref-target".

Simple table attributes

This group includes attributes that are defined only on the `<simpletable>` element: `@keycol` and `@relcolwidth`. These attributes are listed in a group because the `<simpletable>` element is frequently used as a specialization base.

@keycol (simpletable attributes)
Specifies the column that contains the content that represents the key to the tabular structure. If `@keycol` is present and assigned a numerical value, the specified column is treated as a vertical header.

@relcolwidth (simpletable attributes)
Specifies the width of each column in relationship to the width of the other columns. The value is a space-separated list of relative column widths. Each column width is specified as a positive integer or decimal number followed by an asterisk character.

For example, the value `relcolwidth="1* 2* 3*"` gives a total of 6 units across three columns. The relative widths are 1/6, 2/6, and 3/6 (16.7%, 33.3%, and 50%). Similarly, the value `relcolwidth="90* 150*"` causes relative widths of 90/240 and 150/240 (37.5% and 62.5%).

Table accessibility attributes

This group defines a set of attributes that promote table accessibility:

@headers
Specifies which entries in the current table provide headers for this cell. The `@headers` attribute contains an unordered set of unique, space-separated tokens, each of which is an ID reference of an entry from the same table.

@scope
Specifies that the current entry is a header for other table entries. The following values are valid:

col
Indicates that the current entry is a header for all cells in the column.

colgroup
Indicates that the current entry is a header for all cells in the columns that are spanned by this entry.

row
Indicates that the current entry is a header for all cells in the row.

rowgroup
Indicates that the current entry is a header for all cells in the rows that are spanned by this entry.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

Universal attributes

This group defines a set of attributes that are available on almost all DITA elements. It includes all elements in the ID, localization, and metadata attribute groups, as well as the following attributes:

@class *(not for use by authors)*
This attribute is not for use by authors. If an editor displays `@class` *attribute values, do not edit them.* Specifies a default value that defines the specialization ancestry of the element. Its predefined values allow DITA tools to work correctly with specialized elements. In a generalized DITA document the `@class` attribute value in the generalized instance might differ from the default value for the `@class` attribute for the element as given in the DTD or schema. See The class attribute rules and syntax for more information. This attribute is specified on every element except for the `<dita>` container element. It is always specified with a default value, which varies for each element.

@outputclass
Specifies a role that the element is playing. The role must be consistent with the basic semantic and expectations for the element. In particular, the `@outputclass` attribute can be used for styling during output processing; HTML output will typically preserve `@outputclass` for CSS processing.

> **Comment by robander**
> I don't like "The role must be consistent...", that seems like best practice that cannot be normative – and I could easily say outputclass="flashy" which makes my element show up with sparkles, and has nothing to do with "the basic semantic and expectations for the element".

Appendix B.2 Universal attribute group

The universal attribute group defines a set of common attributes that are available on almost every DITA element. The universal attribute group includes all attributes from the ID, localization, and metadata attribute groups, plus the `@class` and `@outputclass` attributes.

> **Comment by Kristen J Eberlein on 29 December 2021**

> This is something wrong with the organizational structure of this topic ... Look at it in outline form, and check that the sections, titles, and content all make logical sense with the topic title of "Universal attribute group".

Common attribute groups

The following attribute groups are referenced in this specification. They are also used in the grammar files when the element attributes are defined.

Universal attributes

Includes `@class` and `@outputclass`, along with every attribute in the ID, localization, and metadata attribute groups.

ID attributes

This group includes the attributes that enable the naming and referencing of elements: `@conaction`, `@conkeyref`, `@conref`, `@conrefend`, and `@id`.

Localization attributes

This group includes attributes that are related to translation and localization: `@dir`, `@translate`, and `@xml:lang`.

Metadata attributes

> **Comment by Kristen J Eberlein on 31 December 2021**
>
> Why do we need to mention that two attributes are available for specialization here? I think it makes the paragraph hard to read.

This group includes common metadata attributes, two of which are available for specialization: `@base`, `@importance`, `@props`, `@rev`, and `@status`.

The base DITA vocabulary from OASIS includes several specializations of `@props`: `@audience`, `@deliveryTarget`, `@otherprops`, `@platform`, and `@product`. These attributes are defined as attribute-extension domains. By default, they are integrated into all OASIS-provided document-type shells, but they can be made unavailable by implementing custom document-type shells.

> **Comment by Kristen J Eberlein on 29 December 2021**
>
> Why do we provide information about specialization and custom document-type shells here? I think that information could be removed.

Universal attribute definitions

The universal attributes for OASIS DITA elements are defined below. Specialized attributes, which are part of the OASIS distribution but are only available when explicitly included in a shell, are noted in the list.

@audience *(specialized attribute)*

Indicates the intended audience for the element. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@base

Specifies metadata about the element. It is often used as a base for specialized attributes that have a simple syntax for values, but which are not conditional processing attributes.

The `@base` attribute takes a space-delimited set of values. However, when serving as a container for generalized attributes, the attribute values will be more complex. See Attribute generalization for more details.

@class *(not for use by authors)*

This attribute is not for use by authors. If an editor displays `@class` *attribute values, do not edit them.* Specifies a default value that defines the specialization ancestry of the element. Its predefined values allow DITA tools to work correctly with specialized elements. In a generalized DITA document the `@class` attribute value in the generalized instance might differ from the default value for the `@class` attribute for the element as given in the DTD or schema. See The class attribute rules and syntax for more information. This attribute is specified on every element except for the `<dita>` container element. It is always specified with a default value, which varies for each element.

@conaction

Specifies how the element content will be pushed into a new location. The following values are valid:

mark

The element acts as a marker when pushing content before or after the target, to help ensure that the push action is valid. The element with `conaction="mark"` also specifies the target of the push action with `@conref`. Content inside of the element with `conaction="mark"` is not pushed to the new location.

pushafter

Content from this element is pushed after the location specified by `@conref` on the element with `conaction="mark"`. The element with `conaction="pushafter"` is the first sibling element after the element with `conaction="mark"`.

pushbefore

Content from this element is pushed before the location specified by `@conref` on the element with `conaction="mark"`. The element with `conaction="pushbefore"` is the first sibling element before the element with `conaction="mark"`.

pushreplace

Content from this element replaces any content from the element referenced by the `@conref` attribute. A second element with `conaction="mark"` is not used when using `conaction="pushreplace"`.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

See STUB CONTENT (56) for examples and details about the syntax.

@conkeyref

Specifies a key name or a key name with an element ID that acts as an indirect reference to reusable content. The referenced content is used in place of the content of the current element. See STUB CONTENT (56) for more details about the syntax and behaviors.

@conref

Specifies a URI that references a DITA element. The referenced content is used in place of the content of the current element. See STUB CONTENT (56) for examples and details about the syntax.

@conrefend

Specifies a URI that references the last element in a sequence of elements, with the first element of the sequence specified by `@conref`. The referenced sequence of elements is used in place of the

content of the current element. See STUB CONTENT (56) for examples and details about the syntax.

@deliveryTarget *(specialized attribute)*
Specifies the intended delivery target of the content, for example, "html", "pdf", or "epub". If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@dir

Identifies or overrides the text directionality. The following values are valid:

lro
Indicates an override of the Unicode Bidirectional Algorithm, forcing the element into left-to-right mode.

ltr
Indicates left-to-right.

rlo
Indicates an override of the Unicode Bidirectional Algorithm, forcing the element into right-to-left mode.

rtl
Indicates right-to-left.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

See The dir attribute for more information.

@id
Specifies an identifier for the current element. This ID is the target for references by `@href` and `@conref` attributes and for external applications that refer to DITA or LwDITA content. This attribute is defined with the XML data type NMTOKEN, except where noted for specific elements within the language reference.

See id attribute for more details.

@importance
Specifies the importance or priority that is assigned to an element. The following values are valid: "default", "deprecated", "high", "low", "normal", "obsolete", "optional", "recommended", "required", "urgent", and "-dita-use-conref-target". This attribute is not used for conditional processing, although applications might use the value of the `@importance` attribute to highlight elements. For example, in steps of a task topic, the value of the `@importance` attribute indicates whether a step is optional or required.

> **Comment by Kristen J Eberlein on 29 September 2022**
>
> I think the phrase "to highlight elements" is a little off. Maybe "render generated text"? And how about adding "Processors often add text or images to ensure that readers of the generated content understand whether the step is optional or required." to the end of the example?

@otherprops *(specialized attribute)*
Specifies a property or properties that provide selection criteria for the element. Alternatively, the `@props` attribute can be specialized to provide a new metadata attribute instead of using the general `@otherprops` attribute. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@outputclass

Specifies a role that the element is playing. The role must be consistent with the basic semantic and expectations for the element. In particular, the `@outputclass` attribute can be used for styling during output processing; HTML output will typically preserve `@outputclass` for CSS processing.

> **Comment by robander**
> I don't like "The role must be consistent...", that seems like best practice that cannot be normative – and I could easily say outputclass="flashy" which makes my element show up with sparkles, and has nothing to do with "the basic semantic and expectations for the element".

@platform *(specialized attribute)*

Indicates operating system and hardware. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

> **Comment by robander**
> I think this could specify a platform that is not an operating system or hardware, right? The current definition explicitly limits platform to those two … maybe "Specifies a platform or platforms to which the element applies, such as the operating system or hardware relevant to a task."

@product *(specialized attribute)*

Specifies the name of the product to which the element applies. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@props

Specifies metadata about the element. New attributes can be specialized from the `@props` attribute. This attribute supports conditional processing. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

The `@props` attribute takes a space-delimited set of values. However, when serving as a container for generalized attributes, the attribute values will be more complex. See Attribute generalization for more details.

@rev

Specifies a revision level of an element that identifies when the element was added or modified. It can be used to flag outputs when it matches a run-time parameter. It cannot be used for filtering nor is it sufficient to be used for version control. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

> **Comment by Kristen J Eberlein on 29 September 2022**
>
> I want to tweak this. How about the following? Also, neither definition describes what values are permitted.
>
> Specifies metadata that identifies when the element was added or the content of the element was modified. The `@rev` attribute can be used for flagging. It cannot be used for filtering nor is it sufficient to be used for version control. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@status
> Specifies the modification status of the element. The following values are valid: "new", "changed", "deleted", "unchanged", and "-dita-use-conref-target".

@translate
> Specifies whether the content of the element should be translated. The following values are valid: "yes", "no", and "-dita-use-conref-target".
>
> See Element-by-element recommendations for translators for suggested processing defaults for each element.
>
> ---
> **Comment by Kristen J Eberlein on 31 December 2021**
>
> Does Element-by-element recommendations for translators really provide suggested processing defaults for each element? I thought it covered whether an element was block or in-line and whether there were considerations that translators needed to be aware of.
> ---

@xml:lang
> Specifies the language and optional locale of the content that is contained in an element. Valid values are language tokens or the null string. The `@xml:lang` attribute and its values are described in the Extensible Markup Language 1.0 specification, fifth edition.
>
> ---
> **Comment by Kristen J Eberlein on 29 September 2022**
>
> Do we also want to direct readers to the architectural topics about the `@xml:lang` attribute?
> ---

Appendix B.3 Common attributes

The common attributes topic collects defines most of the attributes that are used on more than one base element.

Common attribute groups

The following groups are referenced in this specification, and they are also used in grammar files when defining attributes for elements.

Architectural attributes

> This group includes a set of attributes that are defined for document-level elements such as `<topic>` and `<map>`: `@DITAArchVersion`, `@specializations`, and `@xmlns:ditaarch`.

Common map attributes

> This group includes attributes that are frequently used on map elements: `@cascade`, `@chunk`, `@collection-type`, `@keyscope`, `@linking`, `@processing-role`, `@search`, `@toc`, and `@subjectrefs`.

Complex table attributes

> This group includes attributes that are defined on table elements but not simple table elements. These attributes are part of the OASIS Exchange Table Model, unless otherwise noted. Table elements generally use only a subset of the attributes that are defined in this group. This group contains the following attributes: `@align`, `@char`, `@charoff`, `@colsep`, `@rowheader`, `@rowsep`, and `@valign`.

Data-element attributes
Includes attributes defined on `<data>` and its many specializations: `@datatype`, `@name`, and `@value`

Date attributes
Includes attributes that take date values, and are defined on metadata elements that work with date information: `@expiry` and `@golive`

Display attributes

This group includes attributes that affect the rendering of many elements: `@expanse`, `@frame`, and `@scale`.

Inclusion attributes
Includes attributes defined on `<include>` and its specializations: `@encoding` and `@parse`.

Link-relationship attributes

This group includes attributes whose values can be used for representing navigational relationships: `@format`, `@href`, `@type`, and `@scope`.

Simple table attributes

> **Comment by Kristen J Eberlein on 29 December 2021**
>
> If I have jumped to this place in a document from the element-reference topic, I want the attributes listed here in the "Simple table group" to be hyperlinked to the actual definition.

This group includes attributes that are defined only on the `<simpletable>` element: `@keycol` and `@relcolwidth`. These attributes are listed in a group because the `<simpletable>` element is frequently used as a specialization base.

Table accessibility attributes

This group contains attributes that are defined on the `<stentry>` element and its specializations: `@headers` (50) and `@scope` (as defined on `<stentry>`) (55).

Other attributes (not in a group)

These are attributes that are used in the same way on more than one base element, but they are not formally grouped together: `@compact`, `@duplicates`, `@impose-role`, `@otherrole`, `@role`, and `@title-role`.

Common attribute definitions

Common attributes, including those in the groups listed above, are defined as follows.

@align (complex table attributes)
Specifies the horizontal alignment of text in table entries. The following values are valid:

left
Indicates left alignment of the text.

right
Indicates right alignment of the text.

center
Indicates center alignment of the text.

justify
Justifies the contents to both the left and the right.

char
> Indicates character alignment. The text is aligned with the first occurrence of the character specified by the `@char` attribute.

-dita-use-conref-target
> See Using the -dita-use-conref-target value for more information.

The `@align` attribute is available on the following table elements: `<colspec>`, `<entry>`, and `<tgroup>`.

@cascade (common map attributes)

Specifies how metadata attributes cascade within a map. The specification defines the following values:

merge
> Indicates that the metadata attributes cascade, and that the values of the metadata attributes are additive. This is the processing default for the `@cascade` attribute.

nomerge
> Indicates that the metadata attributes cascade, but that they are not additive for `<topicref>` elements that specify a different value for a specific metadata attribute. If the cascading value for an attribute is already merged based on multiple ancestor elements, that merged value continues to cascade until a new value is encountered. That is, setting `cascade="nomerge"` does not undo merging that took place on ancestor elements.

Processors can also define custom, implementation-specific tokens for this attribute.

See Cascading of metadata attributes in a DITA map for more information about how this attribute interacts with metadata attributes.

@char (complex table attributes)
Specifies the alignment character, which is the character that is used for aligning the text in table entries. This attribute applies when `align="char"`. A value of "" (the null string) means there is no aligning character.

For example, if `align="char"` and `char="."` are specified, then text in the table entry aligns with the first occurrence of the period within the entry. This might be useful if decimal alignment is required.

The `@char` attribute is available on the following table elements: `<colspec>` and `<entry>`.

@charoff (complex table attributes)
Specifies the horizontal offset of the alignment character that is specified by the `@char` attribute. The value is a greater-than-zero number that is less than or equal to 100. It represents the percentage of the current column width by which the text is offset to the left of the alignment character.

For example, if `align="char"`, `char="."`, and `charoff="50"` are all specified, then text in the table entry is aligned 50% of the distance to the left of the first occurrence of the period character within the table entry.

The `@charoff` attribute is available on the following table elements: `<colspec>` and `<entry>`.

@chunk (common map attributes)
Specifies how a processor should render a map or branch of a map. For example, it can be used to specify that individual topic documents should be rendered as a single document, or that a single document with multiple topics should be rendered as multiple documents.

The following values are valid:

combine
> Instructs a processor to combine the referenced source documents for rendering purposes. This is intended for cases where a publishing process normally results in a single output artifact for each source XML document.

split
> Instructs a processor to split each topic from the referenced source document into its own document for rendering purposes. This is intended for cases where a publishing process normally results in a single output artifact for each source XML document, regardless of how many DITA topics exist within each source document.

Processors can also define custom, implementation-specific tokens for this attribute.

For a detailed description of the `@chunk` attribute and its usage, see Chunking.

@collection-type (common map attributes)
> Specifies how topics or links relate to each other. The processing default is "unordered", although no default is specified in the OASIS-provided grammar files. The following values are valid:

unordered
> Indicates that the order of the child topics is not significant.

sequence
> Indicates that the order of the child topics is significant. Output processors will typically link between them in order.

choice
> Indicates that one of the children should be selected.

family
> Indicates a tight grouping in which each of the referenced topics not only relates to the current topic but also relate to each other.

Comment by Kristen J Eberlein on 28 September 2022

Here is the content from the "DITA map attributes" topic:

@collection-type
> The `@collection-type` attribute specifies how the children of a `<topicref>` element relate to their parent and to each other. This attribute, which is set on the parent element, typically is used by processors to determine how to generate navigation links in the rendered topics. For example, a `@collection-type` value of "sequence" indicates that children of the specifying `<topicref>` element represent an ordered sequence of topics; processors might add numbers to the list of child topics or generate next/previous links for online presentation. This attribute is available in topics on the `<linklist>` and `<linkpool>` elements, where it has the same behavior. Where the `@collection-type` attribute is available on elements that cannot directly contain elements, the behavior of the attribute is undefined.

Comment by Kristen J Eberlein on 28 September 2022

In the definitions of the supported values, do we want to refer to "resources" instead of "topics"? Since we specify that `@collection-type` specifies "how topics **or links** relate to each other" ...

@colsep (complex table attributes)
Specifies whether to render column separators between table entries. The following values are valid: "0" (no separators) and "1" (separators).

The `@colsep` attribute is available on the following table elements: `<colspec>`, `<entry>`, `<table>`, and `<tgroup>`.

@compact
Specifies whether the vertical spacing between list items is tightened. The following values are valid: "yes", "no", and "-dita-use-conref-target". Some DITA processors or output formats might not support the `@compact` attribute.

@datatype (data-element attributes)
Specifies the type of data contained in the `@value` attribute or within the `<data>` element. A typical use of `@datatype` will be the identifying URI for an XML Schema datatype.

@DITAArchVersion (architectural attributes)
Specifies the version of the DITA architecture that is in use. This attribute is in the namespace `http://dita.oasis-open.org/architecture/2005/`. This attribute is specified in the topic and map modules, and it uses a default value of the current version of DITA. The current default is "2.0".

@duplicates
Specifies whether duplicate links are removed from a group of links. Duplicate links are links that address the same resource using the same properties, such as link text and link role. How duplicate links are determined is processor-specific. The following values are valid:

yes
Specifies that duplicate links are retained.

no
Specifies that duplicate links are removed.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

The suggested processing default is "yes" within `<linklist>` elements and "no" for other links.

> **Comment by robander on Dec 28 2021**
> "How duplicate links are determined is processor-specific" ==> this should be included in any updates to standardize language around "implementation dependent".

@encoding (inclusion attributes)

> **Comment by Kristen J Eberlein on 29 April 2019**
>
> Can we replace "should" in the following definition?

Specifies the character encoding to use when translating the character data from the referenced content. The value should be a valid encoding name. If not specified, processors may make attempts to automatically determine the correct encoding, for example using HTTP headers, through analysis of the binary structure of the referenced data, or the `<?xml?>` processing instruction when including XML as text. The resource should be treated as UTF-8 if no other encoding information can be determined.

When `parse="xml"`, standard XML parsing rules apply for the detection of character encoding. The necessity and uses of `@encoding` for non-standard values of `@parse` are implementation-dependent.

@expanse (display attributes)

Specifies the horizontal placement of the element. The following values are valid:

column

Indicates that the element is aligned with the current column margin.

page

Indicates that the element is placed on the left page margin for left-to-right presentation or the right page margin for right-to-left presentation.

spread

Indicates that the object is rendered across a multi-page spread. If the output format does not have anything that corresponds to spreads, then "spread" has the same meaning as "page".

textline

Indicates that the element is aligned with the left (for left-to-right presentation) or right (for right-to-left presentation) margin of the current text line and takes indentation into account.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

For `<table>`, in place of the `@expanse` attribute that is used by other DITA elements, the `@pgwide` attribute is used in order to conform to the OASIS Exchange Table Model.

Some processors or output formats might not support all values.

@expiry (date attributes)

Specifies the date when the information should be retired or refreshed. The date is specified using the ISO 8601 format: *YYYY-MM-DD*, where *YYYY* is the year, *MM* is the month (01 to 12), and *DD* is the day (01-31).

@format (link-relationship attributes)

Specifies the format of the resource that is referenced. See STUB CONTENT (56) for detailed information on supported values and processing implications.

@frame (display attributes)

Specifies which portion of a border surrounds the element. The following values are valid:

all

Indicates that a line is rendered at the top, bottom, left, and right of the containing element.

bottom

Indicates that a line is rendered at the bottom of the containing element.

none

Indicates that no lines are rendered.

sides

Indicates that a line is rendered at the left and right of the containing element.

top

Indicates that a line is rendered at the top of the containing element.

topbot

Indicates that a line is rendered at the top and bottom of the containing element.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

Some processors or output formats might not support all values.

@golive (date attributes)

Specifies the publication or general availability (GA) date. The date is specified using the ISO 8601 format: *YYYY-MM-DD*, where *YYYY* is the year, *MM* is the month (01 to 12), and *DD* is the day (01-31).

@headers

Specifies which entries in the current table provide headers for this cell. The `@headers` attribute contains an unordered set of unique, space-separated tokens, each of which is an ID reference of an entry from the same table.

@href (link-relationship attributes)

Specifies a reference to a resource. See STUB CONTENT (56) for detailed information on supported values and processing implications.

@impose-role

Specifies whether this element will impose its role on elements in a referenced map. The attribute is ignored if the target of the reference is not a map or branch of a map. The following values are valid:

keeptarget

The role of the current reference is not imposed on the target of the reference. This is the default for the unspecialized `<topicref>` element and for many convenience elements such as `<keydef>`.

impose

The role of the current reference is imposed on the target of the reference. For example, if a specialized topic reference `<chapter>` uses this value and references a map, a topic reference that resolves in place of the `<chapter>` will be treated as if it were a chapter.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

See STUB CONTENT (56) for detailed information on supported values and processing implications.

@keycol (simpletable attributes)

Specifies the column that contains the content that represents the key to the tabular structure. If `@keycol` is present and assigned a numerical value, the specified column is treated as a vertical header.

@keyref

Specifies a key name that acts as a redirectable reference based on a key definition within a map. See STUB CONTENT (56) for information on using this attribute.

For HDITA, the equivalent of `@keyref` is `@data-keyref`

> **Comment by robander**
> The definiton above for @keyref should be synchronized with the definition in the linked section on keys.

@keys

Specifies one or more names for a resource. See STUB CONTENT (56) for information on using this attribute.

For HDITA, the equivalent of `@keys` is `@data-keys`

@keyscope (common map attributes)

Specifies that the element marks the boundaries of a key scope.

See STUB CONTENT (56) for information on using this attribute.

@linking (common map attributes)
Specifies linking characteristics of a topic specific to the location of this reference in a map. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see Cascading of metadata attributes in a DITA map).

The following values are valid:

targetonly
A topic can only be linked to and cannot link to other topics.

sourceonly
A topic cannot be linked to but can link to other topics.

normal
A topic can be linked to and can link to other topics. Use this to override the linking value of a parent topic.

none
A topic cannot be linked to or link to other topics.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

> **linking="sourceonly"**
>> Specifies that the topic will link to its related topics but not vice versa.
>
> **linking="targetonly"**
>> Specifies that the related topics will link to it but not vice versa.
>
> **linking="normal"**
>> Default value. It specifies that linking will be reciprocal (the topic will link to related topics, and they will link back to it).
>
> Authors also can create links directly in a topic by using the `<xref>` or `<link>` elements, but in most cases map-based linking is preferable, because links in topics create dependencies between topics that can hinder reuse.
>
> Note that while the relationships between the topics that are referenced in a map are reciprocal, the relationships merely *imply* reciprocal links in generated output that includes links. The rendered navigation links are a function of the presentation style that is determined by the processor.

@name (data-element attributes)
Defines a unique name for the object.

> **Comment by robander**
> Do we need to specify the scope of "unique" here?

@otherrole
Specifies an alternate role for a link relationship when the `@role` attribute is set to "other".

@parse (inclusion attributes)
Specifies the processing expectations for the referenced resource. Processors must support the following values:

text

The contents should be treated as plain text. Reserved XML characters should be displayed, and not interpreted as XML markup.

xml

The contents of the referenced resource should be treated as an XML document, and the referenced element should be inserted at the location of the `<include>` element. If a fragment identifier is included in the address of the content, processors must select the element with the specified ID. If no fragment identifier is included, the root element of the referenced XML document is selected. Any grammar processing should be performed during resolution, such that default attribute values are explicitly populated. Prolog content must be discarded.

It is an error to use `parse="xml"` anywhere other than within `<foreign>` or a specialization thereof.

Processors may support other values for the `@parse` attribute with proprietary processing semantics. Processors should issue warnings and use `<fallback>` when they encounter unsupported `@parse` values. Non-standard `@parse` instructions should be expressed as URIs.

> **Note** Proprietary `@parse` values will likely limit the portability and interoperability of DITA content, so should be used with care.

@processing-role (common map attributes)

Specifies whether the referenced resource is processed normally or treated as a resource that is only included in order to resolve references, such as key or content references. The following values are valid:

normal

Indicates that the resource is a readable part of the information set. It is included in navigation and search results. This is the default value for the `<topicref>` element.

resource-only

Indicates that the resource should be used only for processing purposes. It is not included in navigation or search results, nor is it rendered as a topic. This is the default value for the `<keydef>` element.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@relcolwidth (simpletable attributes)

Specifies the width of each column in relationship to the width of the other columns. The value is a space-separated list of relative column widths. Each column width is specified as a positive integer or decimal number followed by an asterisk character.

For example, the value `relcolwidth="1* 2* 3*"` gives a total of 6 units across three columns. The relative widths are 1/6, 2/6, and 3/6 (16.7%, 33.3%, and 50%). Similarly, the value `relcolwidth="90* 150*"` causes relative widths of 90/240 and 150/240 (37.5% and 62.5%).

@role

Specifies the role that a linked topic plays in relationship with the current topic.

For example, in a parent/child relationship, the role would be "parent" when the target is the parent of the current topic, and "child" when the target is the child of the current topic. This can be used to sort and classify links when rendering.

The following values are valid:

ancestor

Indicates a link to a topic above the parent topic.

child

Indicates a link to a direct child such as a directly nested or dependent topic.

cousin

Indicates a link to another topic in the same hierarchy that is not a parent, child, sibling, next, or previous.

descendant

Indicates a link to a topic below a child topic.

friend

Indicates a link to a similar topic that is not necessarily part of the same hierarchy.

next

Indicates a link to the next topic in a sequence.

other

Indicates any other kind of relationship or role. The type of role is specified as the value for the `@otherrole` attribute.

parent
> Indicates a link to a topic that is a parent of the current topic.

previous
> Indicates a link to the previous topic in a sequence.

sibling
> Indicates a link between two children of the same parent topic.

-dita-use-conref-target
> See Using the -dita-use-conref-target value for more information.

@rowheader (complex table attributes)

Specifies whether the entries in the respective column are row headers. The following values are valid:

firstcol
> Indicates that entries in the first column of the table are row headers. This applies when the `@rowheader` attribute is specified on the `<table>` element.

headers
> Indicates that entries of the column that is described using the `<colspec>` element are row headers. This applies when the `@rowheader` attribute is specified on the `<colspec>` element.

norowheader
> Indicates that entries in the first column are not row headers. This applies when the `@rowheader` attribute is specified on the `<table>` element.

-dita-use-conref-target
> See Using the -dita-use-conref-target value for more information.

> **Note**   This attribute is not part of the OASIS Exchange Table Model upon which DITA tables are based. Some processors or output formats might not support all values.

The `@rowheader` attribute is available on the following table elements: `<table>` and `<colspec>`.

@rowsep (complex table attributes)

Specifies whether to render row separators between table entries. The following values are valid: "0" (no separators) and "1" (separators).

The `@rowsep` attribute is available on the following table elements: `<colspec>`, `<entry>`, `<row>`, `<table>`, and `<tgroup>`.

@scale (display attributes)

Specifies the percentage by which fonts are resized in relation to the normal text size. The value of this attribute is a positive integer. When used on `<table>` or `<simpletable>`, the following values are valid: "50", "60", "70", "80", "90", "100", "110", "120", "140", "160", "180", "200", and -dita-use-conref-target (56).

This attribute is primarily useful for print-oriented display. Some processors might not support all values.

If the `@scale` attribute is specified on an element that contains an image, the image is not scaled. The image is scaled **only** if a scaling property is explicitly specified for the `<image>` element.

@scope (link-relationship attributes)

Specifies the closeness of the relationship between the current document and the referenced resource. The following values are valid: "local", "peer", "external", and "-dita-use-conref-target".

See STUB CONTENT (56) for detailed information on supported values and processing implications.

@scope
Specifies that the current entry is a header for other table entries. The following values are valid:

col
Indicates that the current entry is a header for all cells in the column.

colgroup
Indicates that the current entry is a header for all cells in the columns that are spanned by this entry.

row
Indicates that the current entry is a header for all cells in the row.

rowgroup
Indicates that the current entry is a header for all cells in the rows that are spanned by this entry.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

@search (common map attributes)
Specifies whether the target is available for searching. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see Cascading of metadata attributes in a DITA map). The following values are valid: "yes", "no", and "-dita-use-conref-target".

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> Here is the content from the "DITA map attributes" topic:
>
> **@search**
> Specifies whether the topic is included in search indexes.

@specializations (architectural attributes)
Specifies the attribute-domain specializations that are included in the document-type shell. This attribute is set as a default within the document-type shell. The value varies depending on what domains are integrated into the document-type shell. For example, a grammar file that includes the specialized attributes `@audience`, `@deliveryTarget`, and `@newBaseAtt` would set the value to `@props/audience @props/deliveryTarget @base/newBaseAtt`.

@subjectrefs (common map attributes)
Specifies one or more keys that are each defined by a subject definition in a subject scheme map. Multiple values are separated by white space.

@title-role (REQUIRED)
Specifies the role that the alternative title serves. Multiple roles are separated by white space. The following roles are defined in the specification: "linking", "navigation", "search", "subtitle", and "hint".

Processors can define custom values for the `@title-role` attribute.

@toc (common map attributes)
Specifies whether a topic appears in the table of contents (TOC) based on the current map context. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see Cascading of metadata attributes in a DITA map). The following values are valid:

yes
The topic appears in a generated TOC.

no
The topic does not appear in a generated TOC.

-dita-use-conref-target
> See STUB CONTENT (56) for more information.

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> Here is the content from the "DITA map attributes" topic:
>
> **@toc**
>> Specifies whether topics are excluded from navigation output, such as a Web site map or an online table of contents. By default, `<topicref>` hierarchies are included in navigation output; relationship tables are excluded.

@type (link-relationship attributes)
> Describes the target of a reference. See STUB CONTENT (56) for detailed information on supported values and processing implications.

@value (data-element attributes)
> Specifies a value associated with the current property or element.

@valign (complex table attributes)
> Specifies the vertical alignment of text in table entries. The following values are valid:

> **bottom**
>> Indicates that text is aligned with the bottom of the table entry.

> **middle**
>> Indicates that text is aligned with the middle of the table entry.

> **top**
>> Indicates that text is aligned with the top of the table entry.

> **-dita-use-conref-target**
>> See Using the -dita-use-conref-target value for more information.

> The `@valign` attribute is available on the following table elements: `<entry>`, `<tbody>`, `<thead>`, and `<row>`.

@xml:space
> Specifies how to handle white space in the current element. This attribute is provided on `<pre>`, `<lines>`, and on elements specialized from those. It ensures that parsers respect white space that is part of the data in those elements, including line-end characters. When defined, it has a fixed value of "preserve", making it a default property of the element that cannot be changed or deleted by authors.

@xmlns:ditaarch (architectural attributes)
> Declares the default DITA namespace. This namespace is declared as such in the RNG modules for `<topic>` and `<map>`, but it is specified as an attribute in the equivalent DTD-based modules. The value is fixed to "http://dita.oasis-open.org/architecture/2005/".

Appendix B.4 STUB CONTENT

STUB CONTENT
> STUB CONTENT

STUB CONTENT
> STUB CONTENT

STUB CONTENT
> STUB CONTENT

STUB CONTENT
　　STUB CONTENT

STUB CONTENT
　　STUB CONTENT

STUB CONTENT
　　STUB CONTENT

STUB CONTENT
　　STUB CONTENT

STUB CONTENT
　　STUB CONTENT

STUB CONTENT
　　STUB CONTENT

STUB CONTENT
　　STUB CONTENT

STUB CONTENT
　　STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT (58)

STUB CONTENT (58)

```
STUB CONTENT
```

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

Index

A

B

C

D

F

G

I

M

N

P

T

U