

Review B: Concept and reference

Table of contents

1 Topic and map document types.....	3
1.1 Concept.....	3
1.2 Reference.....	5
2 Concept elements.....	8
2.1 <concept>.....	8
2.2 <conbody>.....	9
2.3 <conbodydiv>.....	9
3 Reference elements.....	12
3.1 <propdesc>.....	12
3.2 <propdeschd>.....	12
3.3 <properties>.....	13
3.4 <property>.....	15
3.5 <prophead>.....	16
3.6 <proptype>.....	17
3.7 <proptypehd>.....	18
3.8 <propvalue>.....	18
3.9 <propvaluehd>.....	19
3.10 <refbody>.....	19
3.11 <refbodydiv>.....	20
3.12 <reference>.....	20
3.13 <refsyn>.....	22
A Aggregated RFC-2119 statements.....	24
B Attributes.....	25
B.1 Attribute groups.....	25
B.2 Universal attribute group.....	38
B.3 Common attributes.....	43
B.4 STUB CONTENT.....	55
C Formatting expectations.....	57
Index.....	59

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 Concept

Concept topics are designed to provide conceptual or descriptive information.

Purpose

Concept topics serve a variety of purposes:

- Provide background information that helps readers understand essential facts about a product, process, or task
- Provide an extended definition of a major abstraction, such as a process or function
- Explain the nature and components of a product and describe how it fits into a category of products
- Help readers map their knowledge and understanding to the tasks that they need to perform

Content model

The body of a concept topic can contain the following document structures:

- Basic document structures: divisions, paragraphs, lists, tables, figures, etc.
- Concept body divisions: `<conbodydiv>` (9)
- Examples
- Sections

However, after a section or example is introduced into the topic structure, it can be followed only by a section, example, or concept body division.

Comment by Kristen J Eberlein on 07 October 2022

What is rationale for this design choice? It would be good to explain it

Zoë Lawson 17 Nov 2022

I wasn't around when the decision was made, but does it have to do with it being uncommon to return to a 'parent' content without some sort of title? If I'm working with a document, if I'm in section 1.3.4, I don't just "go back" into section 1.3, I'm either going into section 1.3.5, or 1.4.

The only use case I can think of where I might want to go back would be if I were trying to make a sidebar, which I guess could be a common use case for example. DITA doesn't really have a 'sidebar' concept without specialization, that I can think of.

Kris Eberlein, 18 November 2022

I e-mailed with Michael Priestley today. Based on our e-mail exchange, I added the following content (and removed this draft comment:

"This design supports the following best practices and use cases:

- Ensures that there are clear boundaries between sections
- Enables creation of short concept topics that only include a few basic block elements, for example, a paragraph, list, and image"

Disposition: Completed

Example

Comment by Kristen J Eberlein on 07 October 2022

I think the content of the <example> section sucks. For the examples in this "Overview" section, I think it's important that they be solid topics.

Zoë 15 Nov

Would a topic about a math or science concept work? Of course, I can't think of one that isn't straight out of the guide I'm currently working on that I can't share. Random ideas:

- What is a triad color scheme, and an example (although that might need an image, unless you name the colors in a color wheel).
- Defining an exothermic reaction, and some chemical examples?
- Ohm's law?
- Classifying different types of mead, and the example of how cyser is a melomel but not all melomels are cysers?

Kris Eberlein, 16 November 2022

I think the triad color scheme might be a good example. Are you game to work it up? I think the other ideas are not going to be intuitive enough.

Zoë Lawson, 17 Nov 2022

I tried. Short topics are hard. And content fusion does not like to paste in xml, even in a codeblock.

```
<concept id="color_triads">
<title>Color Triads</title>
<shortdesc>A basic concept in color theory is the use of triads, colors that are three
steps apart on the color wheel.</shortdesc>
<conbody>
  <p>If you have difficulty picking colors when designing, you can use a color triad. Use
one color as the main color and then the other two as accent colors.</p>
<example>
  <p>The most simple color wheel contains 12 colors: red, red-orange, orange, yellow-
orange, yellow, yellow-green, green, blue-green, blue, blue-violet, violet (purple), and
red-violet.</p>
  <fig>
    <title>Basic color wheel</title>
    <image href="colorwheel.jpg" placement="break"><alt>Circle divided into twelve
parts, each part a different color.</alt></image>
  </fig>
  <p>Start with the first color green. Skip the next three colors, and the second color
of the triad is violet. Skip the next three colors, and the third color of the triad is
orange. This gives you the triad of green, violet, and orange.</p>
</example>
</conbody>
</concept>
```

Kris Eberlein, 18 November 2022

Thanks; I've added your example (and got rid of the previous one).

Disposition: Completed

The following code sample contains a simple concept topic:

```
<concept id="concept">
  <title>Bird Songs</title>
  <shortdesc>Bird songs are complex vocalizations used to attract mates or defend
    territories.</shortdesc>
  <conbody>
    <p>Bird songs vary widely among species, from simple songs that are genetically
      imprinted to complex songs that are learned over a lifetime.</p>
    <example>
      <p>Flycatchers know their songs from birth:</p>
      <ul>
        <li>Flycatcher songs are simple sequences of notes.</li>
        <li>Flycatcher songs never vary but are unique to each member of the Flycatcher
          family.</li>
      </ul>
    </example>
  </conbody>
</concept>
```

1.2 Reference

Reference topics contain reference information that users might need to consult occasionally, for example, product specifications, part lists, API calls, and programming language commands.

Purpose and usage

Reference topics serve the following purposes:

- Provide data that supports novice users as they perform a task
- Provide quick access to fact-based information
- Contain information that advanced users look up infrequently

Comment by Zoë Lawson on 15 Nov

Do we really want to split between data for novice users vs advanced users? I'm not sure that distinction is necessary. I think just "...supports users as they perform a task" and "...information that users look up infrequently" are perfectly fine.

Kris Eberlein, 17 November 2022

Hmm ... This has been a distinction since the very early days of DITA. Also, it would leave us with only two list items. Can you think of other purpose and usage items for reference topics?

Zoë Lawson, 17 Nov 202

Why does removing the user type/audience change the number of list items? I think the three items are just as valid. But it's hard to be verbose about reference content. It's stuff you reference. I consider myself something of a not-novice cook, but I have to re-read a recipe when I'm baking dozens of times to confirm I haven't confused baking powder vs baking soda. Reference material is for everyone. It's written down, so I don't have to spend brain cells on memorizing stuff.

But, I'm not super concerned about it. If you want to keep the audience words, that's fine.

Kris Eberlein, 18 November 2022

Changed "novice users" to "users". Changed the third list item to read "Contain detailed information that users look up infrequently"

Disposition: Completed

Reference topics are used for the following types of information and more:

- API documentation
- Bibliographies
- Configuration file options
- Catalogs
- Element references
- Lists of equipment, ingredients, parts, and tools
- Specifications
- Syntax diagrams and explanations

Content model

The body of a reference topic can contain the following document structures:

- Examples: `<example>`
- Property lists: `<properties>` (13)
- Reference body divisions: `<refbodydiv>` (20)
- Sections: `<section>`
- Syntax sections: `<refsyn>` (22)
- Tables: `<simpletable>` and `<table>`

These structures can appear in any order or combination. However, basic document structure such as paragraphs, lists, and figures cannot be placed directly in the body of the reference topic. They must be contained within one of the structures listed above.

Example

The following code sample contains a simple reference topic:

Comment by Kristen J Eberlein on 26 October 2022

While the following code sample is a perfectly reasonable example of a properties list, I think we could provide a better example of a reference topic.

Zoe 15 Nov

I don't know if we really want to show a syntax diagram, but possibly a command that can take a set of parameters, where you have the command syntax in a `<refsyn>` and then the parameters in a properties table?

Disposition: Unassigned

```
<reference id="oil-types">
  <title>Oil types</title>
  <shortdesc>The tables provide the recommended oil types.</shortdesc>
  <refbody>
    <properties>
      <prophead>
        <proptypehd>Oil type</proptypehd>
        <propvaluehd>Oil brand</propvaluehd>
```

```
<propdesch>Appropriate use</propdesch>
</prophead>
<property>
  <proptype>Primary oil</proptype>
  <propvalue>A1X</propvalue>
  <propdesc>One-cylinder engines</propdesc>
</property>
<property>
  <proptype>Secondary oil</proptype>
  <propvalue>B2Z</propvalue>
  <propdesc>Two-cylinder engines</propdesc>
</property>
</properties>
</refbody>
</reference>
```

2 Concept elements

Concept elements provide the fundamental structure for concept topics. Concept topics are useful for introducing the background or overview information for task or reference topics.

2.1 <concept>

The <concept> element is the top-level element for a topic that answers the question “what is?”

Usage information

Concepts provide background information that users must know before they can successfully work with a product or interface. Often, a concept is an extended definition of a major abstraction such as a process or function. It might also have an example or a graphic.

Comment by Zoë Lawson on 15 Nov

Instead of "graphic" maybe "diagram"? My brain equates diagram with flowcharts, examples, etc. faster than "graphic".

Also, do we want to keep it singular? "It might also contain examples or diagrams." By using singular, that might imply that you can only have one, when that's not true.

Kris Eberlein, 18 November 2022

Changed to "It might also contain an example, image, or diagram." I kept it singular because I think it reads better that way.

Disposition: Completed

Specialization hierarchy

The <concept> element is specialized from <topic>. It is defined in the concept module.

Attributes

The following attributes are available on this element: [universal attributes \(38\)](#) and [architectural attributes \(43\)](#).

For this element, the @id attribute is required.

Example

The following code sample shows a concept topic:

```
<concept id="concept">
  <title>DITA concept topic</title>
  <shortdesc>The concept topic answers the question <q>what is?</q></shortdesc>
  <conbody>
    <p>Concept topics provide background information that users must know
      before they can successfully work with a product or interface. Often,
      a concept is an extended definition of a major abstraction such as a
      process or function. It might also have an example or a graphic.</p>
  </conbody>
</concept>
```


2.2 <conbody>

The <conbody> element contains the main content of a concept topic.

Usage information

The <conbody> element allows paragraphs, lists, and other elements as well as sections and examples. However, <conbody> element has a restriction that a <section> or an <example> can be followed only by other sections, examples, or <conbodydiv> elements that group sections and examples.

Specialization hierarchy

The <conbody> element is specialized from <body>. It is defined in the concept module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38).

Example

See [<concept>](#) (8).

2.3 <conbodydiv>

The <conbodydiv> element provides an container for content that might be grouped within a concept topic.

Usage information

There are no additional semantics attached to the <conbodydiv> element. It is purely a grouping element that provided to help organize content for reuse.

Comment by Zoë Lawson on 15 Nov

"It is purely a grouping element provided to help organize content for reuse." or "It is purely a grouping element that is provided to help organize content for reuse." Drop the "that" or add "that is"

Kris Eberlein, 16 November 2022

Done.

Disposition: Completed

The parent <conbody> element has a restriction that <section> or <example> elements can only be followed by other sections or examples. The <conbodydiv> element, which allows groupings of sections and examples, keeps the same restriction in place; once used, only sections, examples, or other <conbodydiv> groups are allowed.

Comment by Eliot Kimber

Would it be useful to point out that <div> can be used for grouping of elements that occur before any section, example, or conbodydiv?

Kris Eberlein, 15 November 2022

I think that's a little too specific. I think that would be appropriate for training materials about reuse, not the spec.

Zoë Lawson 15 Nov 2022

My brain got spun around by the paragraph. Beginning the paragraph with only listing the `<section>` and `<example>` confused me.

`<conbodydiv>` exists so you can either wrap sections and examples up for reuse OR to make another chunk after a `<section>` or `<example>` that does not include a title? Is the purpose just to provide a grouping mechanism, and just because of how all the specializations work, you can also use it after a `<section>` or `<example>`?

Kris Eberlein, 18 November 2022

@Zoe, `<conbodydiv>` exists so that one can group sections and examples together for reuse. Its purpose is not to "make another chunk after a `<section>` or `<example>` that does not include a title" – although one could always have a `<section>` or `<example>` within a `<conbodydiv>` that does not have a title. (Titles are always optional in `<section>` and `<example>`.)

It's clear that you were thrown by the paragraph that begins "The parent `<conbody>` element ...". That's legacy content, and I don't know if it is even needed. Here's what I think it was trying to communicate:

- The `<conbody>` element has a content model that allows basic block elements but ONLY until a section or example or `<conbodydiv>` is inserted. After a section or example or `<conbodydiv>` is inserted, then only other sections or examples or `<conbodydiv>` elements can follow.
- The content model of `<conbodydiv>` only allows `<section>` or `<example>`.

I've replaced the troublesome paragraph with a simple sentence: "The content model of `<conbodydiv>` only includes `<section>` and `<example>`."

Disposition: Completed

Specialization hierarchy

The `<conbodydiv>` element is specialized from `<bodydiv>`. It is defined in the concept module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38).

Example

The following code sample shows how a `<conbodydiv>` element can be used to group content for reuse:

```
<conbody>
  <conbodydiv id="concept-purpose-content-model">
    <section id="purpose">
      <title>Purpose</title>
      <p>Concept topics serve a variety of purposes:</p>
      <!-- ... -->
    </section>
    <section id="content-model">
      <title>Content model</title>
      <p>The body of a concept topic can contain the following document
        structures:</p>
      <!-- ... -->
    </section>
  </conbodydiv>
</conbody>
```

```
</section>  
</conbodydiv>  
</conbody>
```

3 Reference elements

Reference elements provide the fundamental structure for reference topics. Reference topics include specialized sections for programming language syntax and property lists, as well as standard elements such as sections, tables, and examples.

3.1 <propdesc>

The <propdesc> element contains the content of the third column in a properties table. This content describes the property type and its values.

Comment by Eliot Kimber

contains the description of the property described by a property list item.

The position of the element in the table is not relevant to the element's semantics.

Kris Eberlein, 15 November 2022

How about simply "The <propdesc> element contains content that describes the property type and its values."?

Kris Eberlein, 18 November 2022

This is ugly. While we know that this element is designed to provide descriptive content, we don't know whether what is described is the "type" or the "value".

None the less, changed as I suggested above.

Disposition: Completed

Specialization hierarchy

The <propdesc> element is specialized from <stentry>. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38), [table accessibility attributes](#) (44), and the attributes defined below.

@rowspan

Specifies the number of rows that a **cell** is to span inside a simple table.

Example

See [<properties>](#) (13).

3.2 <propdeschd>

The <propdeschd> element provides a label for the description column in a properties table.

Specialization hierarchy

The <propdeschd> element is specialized from <stentry>. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38) and [table accessibility attributes](#) (44).

Example

See `<properties>` (13).

3.3 `<properties>`

A properties table describes the properties of something. Each property can include the type, value, and a description.

Comment by Kristen J Eberlein on 26 October 2022

Can we be more specific than "something"? Perhaps "an object, category, or X". Thoughts?

Eliot Kimber: I would probably say "a thing, such as an object or part."

Kris Eberlein, 18 November 2022

Changed to "A properties table describes the properties of a thing, such as an object, part, or category. Each property can include the type, value, and a description."

Disposition: Completed

Comment by Zoë Lawson on 15 November 2022

Do we need to explain what "type", "value", and "description" could be?

I may be confused by half-remembered IBM-isms. I feel like once upon a time it was explained to me that "proptype" was the 'name', "propvalue" was a default value, and then the "propdesc" was the description.

Kris Eberlein, 16 November 2022

I don't think we need to define what "type", "value", and "description" are.

Disposition: Closed

Usage information

A properties table typically is represented as a simple table with three columns. The first cell in a row lists the property type, the second cell in the row lists a value for the property, and the third cell in the row lists a description,

An optional header row can provide labels for the columns, if an author does not want to use the default labels that might be provided by stylesheets.

Comment by Zoë Lawson on 15 Nov 2022

I have a vague memory that the three columns weren't required, that you could have a two column table. Was that an IBM-ism, or is it technically possible? If you can do two columns, do we describe that more specifically?

Kris Eberlein, 16 November 2022

I was about to say authoritatively that a two-column properties table was not permitted, but then I thought I better test. Yes, you can have a two-column – or even a one-column properties table. I think this is a useless element.

Kris Eberlein, 18 November 2022

Changed to read "A properties table typically is represented as a simple table with a maximum of three columns. The first column is for the property type, the second column can contain a value for the property, and the third column can contain a description."

Disposition: Completed

Specialization hierarchy

The `<properties>` element is specialized from `<simpletable>`. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38), [display attributes](#) (44), and [simpletable attributes](#) (44).

Examples

This section contains examples of how the `<properties>` element can be used.

Figure 1: Simple properties list

Comment by Zoë Lawson on 15 Nov 2022

Can we please be consistent with "table" over "list"?

ETA: Saw your comment to Eliot in a different topic. I'll stop marking these.

Kris Eberlein, 18 November 2022

Done.

Disposition: Completed

The following code sample shows a `<properties>` element that describes information about motor oil types:

```
<properties>
  <prophead>
    <proptypehd>Oil type</proptypehd>
    <propvaluehd>Oil brand</propvaluehd>
    <propdeschd>Appropriate use</propdeschd>
  </prophead>
  <property>
    <proptype>Primary oil</proptype>
    <propvalue>AlX</propvalue>
    <propdesc>One-cylinder engines</propdesc>
  </property>
  <property>
    <proptype>Secondary oil</proptype>
    <propvalue>B2Z</propvalue>
    <propdesc>Two-cylinder engines</propdesc>
  </property>
</properties>
```

The properties list might be rendered as follows:

Comment by Zoë Lawson on 15 Nov 2022
table instead of list, please.

Kris Eberlein, 18 November 2022

Done.

Disposition: Completed

Oil type	Oil brand	Appropriate use
Primary oil	A1X	One-cylinder engine
Secondary oil	B2X	Two cylinder engine

Figure 2: Properties table with spanned cells

The following code sample shows a properties table with spanned cells:

```
<properties>
  <prophead>
    <proptypehd>Visual element</proptypehd>
    <propvaluehd>Value</propvaluehd>
    <propdeschd>What it does</propdeschd>
  </prophead>
  <property>
    <proptype>Color</proptype>
    <propvalue>Red</propvalue>
    <propdesc>Indicates an error</propdesc>
  </property>
  <property>
    <propvalue>Green</propvalue>
    <propdesc>Indicates that conditions are good</propdesc>
  </property>
  <property>
    <propvalue>Yellow</propvalue>
    <propdesc>Indicates that a problem might exist</propdesc>
  </property>
  <property>
    <proptype>Shape</proptype>
    <propvalue>Circle, square, or triangle</propvalue>
    <propdesc>Use to add contrast and depth</propdesc>
  </property>
</properties>
```

The properties list might be rendered as follows:

Visual element	Value	What it does
Color	Red	Indicates an error
	Green	Indicates that conditions are good
	Yellow	Indicates that a problem might exist
Shape	Circle, square, or triangle	Adds contrast and depth

3.4 <property>

The <property> element represents a single property in a properties list.

Comment by Eliot Kimber

c/properties list/properties table/ or change "properties table" to "properties list" throughout (which I think I would prefer so as to move away from using "table").

Kris Eberlein, 15 November 2022

Changed to "properties table". That's the usage that Robert and I agreed on, especially since some of the attribute definitions are from the table accessibility group and so refer to tables.

Disposition: Completed

Specialization hierarchy

The `<property>` element is specialized from `<strow>`. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38).

Example

See `<properties>` (13).

3.5 `<prophead>`

The `<prophead>` element contains elements that provide labels for the columns in a properties table.

Comment by Eliot Kimber

c/contains elements that provide/provides/

Kris Eberlein, 15 November 2022

No – "provide labels" is correct. The reference is to "elements," "not the `<prophead>` element".

Disposition: Rejected

Rendering expectations

Comment by Kristen J Eberlein on 26 October 2022

What are our expectations if a properties list does not contain a header row? Do we expect stylesheets might generate headings/labels? Do we expect that author-supplied headings always override those provided by stylesheets?

Eliot Kimber

I would expect to have headers generated if not specified and would expect authored headers to take precedence as a general implementation principle ("authored stuff wins").

Kris Eberlein, 15 November 2022

Robert and I discussed this before I opened the review, but I neglected to update the topic. We agree with you, and also wanted to make the rendering expectations parallel with `<choicetable>`.

Made this section parallel with <choicetable>.

Disposition: Completed

Specialization hierarchy

The <prophead> element is specialized from <sthead>. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38).

Example

See <properties> (13).

3.6 <proptype>

The <proptype> element contains the content of the first cell in a properties table row. The content describes the type of the property.

Comment by Eliot Kimber

If we were to move away from using table for what <properties> represents I would make this "Specifies the type of the property described by a property list item."

Kris Eberlein, 15 November 2022

How about "The <proptype> element contains content that describes the type of the property."?

Kris Eberlein, 18 November 2022

Done.

Disposition: Completed

Specialization hierarchy

The <proptype> element is specialized from <stentry>. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38), [table accessibility attributes](#) (44), and the attributes defined below.

@rowspan

Specifies the number of rows that a **cell** is to span inside a simple table.

Example

See <properties> (13).

3.7 <proptyped>

The <proptyped> element provides a label for the type column in a properties table.

Specialization hierarchy

The <proptyped> element is specialized from <stentry>. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38) and [table accessibility attributes](#) (44).

Example

See [<properties>](#) (13).

3.8 <propvalue>

The <propvalue> element contains the content of the second cell in a properties table row. This content indicates a value for the property type.

Comment by Eliot Kimber

specifies the value of the property described by a property list item.

The fact that it's the second item is not really relevant to the element's semantic.

Kris Eberlein, 15 November 2022

How about "The <propvalue> element contains content that indicates a value for the property type."

Disposition: Completed

Usage information

Values can be placed in separate <property> elements if they need separate descriptions. The <proptype> element need not be repeated.

Comment by robander on 20 May 2021

This feels like something that should be called out for processors.

Kris Eberlein, 26 October 2022

We currently also call out the following nuance in the <properties> topic:

"When a single property type takes multiple values, authors can create additional property elements and use only the <propvalue> and <propdesc> elements for each successive value."

Zoë 15 Nov

I'm assuming you removed this paragraph based on the removal of the hack (re: 1 Nov mtg)? Or replace it with reference to the spanned rows example?

Kris Eberlein, 18 November 2022

Yes, I've removed the section.

Disposition: Completed

Specialization hierarchy

The `<propvalue>` element is specialized from `<stentry>`. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38), [table accessibility attributes](#) (44), and the attributes defined below.

@rowspan

Specifies the number of rows that a **cell** is to span inside a simple table.

Example

See `<properties>` (13).

3.9 `<propvaluehd>`

The `<propvaluehd>` element provides a label for the value column in a properties table.

Specialization hierarchy

The `<propvaluehd>` element is specialized from `<stentry>`. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38) and [table accessibility attributes](#) (44).

Example

See `<properties>` (13).

3.10 `<refbody>`

The `<refbody>` element contains the main content of a reference topic.

Specialization hierarchy

The `<refbody>` element is specialized from `<body>`. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38).

Example

See `<reference>` (20).

3.11 <refbodydiv>

The <refbodydiv> element provides a container for contiguous content in a reference topic. There is no additional semantic meaning.

Usage information

The <refbodydiv> element is useful primarily for reuse and as a specialization base.

Specialization hierarchy

The <refbodydiv> element is specialized from <bodydiv>. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38).

Example

The following code sample shows how a <refbodydiv> element can be used to group content for reuse

```
<reference id="sample-refbodydiv" xml:lang="en">
  <title>Sample for refbody</title>
  <shortdesc>This shows how refbodydiv might be used.</shortdesc>
  <refbody>
    <refbodydiv id="widget1">
      <section>This is one part of the sample</section>
      <refsyn>Syntax for this part</refsyn>
    </refbodydiv>
    <refbodydiv id="widget2">
      <section>This is another part of the sample</section>
      <refsyn>Syntax for this part</refsyn>
    </refbodydiv>
  </refbody>
</reference>
```

Comment by Kristen J Eberlein on 26 October 2022

Can someone come up with a more realistic example?

Zoë 15 Nov

Would my earlier idea of a <refsyn> followed by a <properties> defining options in the command work? You'd want the two grouped together for reuse. Can possibly cobble something together from the DITA-OT command line? e.g. args.rellinks has four possible values

Kris Eberlein, 16 November 2022

A <refsyn> element can contain properties, so there is no need for <refbodydiv>. Maybe <refbodydiv> used simply to group two <section> elements?

Disposition: Unassigned

3.12 <reference>

The <reference> element is the top-level element for a reference topic. The reference topic includes special sections for programming syntax and property lists, as well as sections, tables, and examples.

Comment by Kristen James Eberlein on 15 November 2022

The 2nd sentence in the short description needs to be edited. I suggest "A reference topic can include **specialized** sections for programming syntax and property **tables**, as well as **standard** sections, tables, and examples." Bold added to indicate the changes.

Disposition: Completed

Comment by Eliot Kimber

I would say either "Reference topics" or "A reference topic". "The reference topic" seems incorrect.

Kris Eberlein, 15 November 2022

Changed to "A reference topic ..."

Disposition: Completed

Comment by Eliot Kimber

Update to what I wrote below: I see that the Reference topic under Topic and map document types provides the more general guidance I was expecting here. Should probably have a cross reference to that topic from here so.

Original comment as written before finishing my review:

I would expect a bit more usage information about how reference topics are always and only organized into sections and are intended to use the same sets of sections for all the reference entries for a given kind of thing being described. But I don't know if that level of usage information is necessary or appropriate for this spec.

But the second sentence in the shortdesc only hints at a deeper intent for <reference> without really illuminating it at all. In particular, the mention of "special sections" without also mentioning unspecialized sections is bit of a non-sequitur. As we don't reflect the content model in these topics, it leaves a bit of a gap.

So probably necessary to have some enumeration of the base content model.

Kris Eberlein, 15 November 2022

A couple of points:

- Reference topics are NOT always organized into sections, as I do not think we can consider tables to be sections.
- Nor are reference topics "intended to use the same set of sections" – I think that's a choice that an implementation makes or a best practice.

I think that rather than enumerating the content model, what we should do is add a "Usage information" section that contains the following content: "For information about the purpose and content model of reference topics, see <cross reference to the overview topic about reference topics>."

Disposition: Completed

Specialization hierarchy

The <reference> element is specialized from <topic>. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes](#) (38) and [architectural attributes](#) (43).

For this element, the @id attribute is required.

Example

Comment by Kristen J Eberlein on 08 November 2022

This is a pretty lousy example ...

Disposition: Unassigned

The following code sample shows how a reference topic can be used:

```
<reference id="requiredTools">
  <title>Tools required to maintain Acme machinery</title>
  <refbody>
    <section>
      <title>Small tools</title>
      <ul>
        <li>Hard hat</li>
        <li>Hammer</li>
        <li>Nail</li>
        <li>Metal polish</li>
        <!-- .... -->
      </ul>
    </section>
    <section>
      <title>Expensive tools</title>
      <!-- .... -->
    </section>
  </refbody>
</reference>
```

3.13 <refsyn>

The <refsyn> element describes the syntax of something, for example, a command-line utility or an API signature.

Comment by Kris Eberlein on 26 October 2022

Can we be more specific than "something"?

Eliot Kimber: I would never use refsyn for anything other than command line commands or maybe as a type of BNF for something you type (i.e., YAML or something you might describe the syntax of).

So I would say something "the syntax of a command line command or similar typed construct"

Kris Eberlein, 15 November 2022

Changed to "contains content that describes the syntax of a command".

Disposition: Completed

Specialization hierarchy

The <refsyn> element is specialized from <section>. It is defined in the reference module.

Attributes

The following attributes are available on this element: [universal attributes \(38\)](#).

Example

The following code sample shows how the `<refsyn>` element can be used to document the syntax for the Windows `mkdir` command:

```
<refsyn>
  <title>Syntax</title>
  <codeblock>mkdir <varname>drive</varname> <varname>directory</varname></codeblock>
  <parml>
    <plentry>
      <pt><varname>drive</varname></pt>
      <pd>Specifies the drive on which the new directory is created. This is an optional
        parameter.</pd>
    </plentry>
    <plentry>
      <pt><varname>path</varname></pt>
      <pd>Specifies the fully-qualified name of the new directory. This is a required
        parameter.</pd>
    </plentry>
  </parml>
</refsyn>
```

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.

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.

Disposition: Unassigned

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.

Disposition: Unassigned

@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 or 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.

Disposition: Unassigned

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" ...

Disposition: Unassigned

@keyscope (common map attributes)

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

See [B.4 STUB CONTENT](#) (55) 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](#).

Disposition: Unassigned

@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".

Disposition: Unassigned

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.

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.

Disposition: Unassigned**@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.

Disposition: Unassigned

@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 [B.4 STUB CONTENT](#) (55) 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.

Disposition: Unassigned

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?

Disposition: Unassigned

@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.

@expansion (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 `@expansion` 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](#) (55).

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 [B.4 STUB CONTENT](#) (55) 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 [B.4 STUB CONTENT](#) (55) 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 [B.4 STUB CONTENT](#) (55) 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 [B.4 STUB CONTENT](#) (55) 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 ...

Disposition: Unassigned

@encoding (inclusion attributes)

Comment by Kristen J Eberlein on 29 April 2019

Can we replace "should" in the following definition?

Disposition: Unassigned

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 [B.4 STUB CONTENT \(55\)](#) for detailed information on supported values and processing implications.

@href (link-relationship attributes)

Specifies a reference to a resource. See [B.4 STUB CONTENT \(55\)](#) 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 [B.4 STUB CONTENT \(55\)](#) for detailed information on supported values and processing implications.

@type (link-relationship attributes)

Describes the target of a reference. See [B.4 STUB CONTENT \(55\)](#) 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.

Disposition: Unassigned

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.

Disposition: Unassigned

@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?

Disposition: Unassigned

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?

Disposition: Unassigned

@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.

Disposition: Unassigned

@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%).

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".

Disposition: Unassigned

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".

Disposition: Unassigned

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.

Disposition: Unassigned

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.

Disposition: Unassigned

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 [B.4 STUB CONTENT](#) (55) 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 [B.4 STUB CONTENT](#) (55) 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 [B.4 STUB CONTENT](#) (55) 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 [B.4 STUB CONTENT](#) (55) 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?

Disposition: Unassigned

@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".

Disposition: Unassigned

@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."

Disposition: Unassigned

@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.

Disposition: Unassigned

@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.

Disposition: Unassigned

@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?

Disposition: Unassigned

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: `@expand`, `@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.

Disposition: Unassigned

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` (49) and `@scope` (as defined on `<stentry>`) (53).

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`, `@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.

Disposition: Unassigned

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" ...

Disposition: Unassigned

@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".

Disposition: Unassigned

@encoding (inclusion attributes)

Comment by Kristen J Eberlein on 29 April 2019

Can we replace "should" in the following definition?

Disposition: Unassigned

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.

@expand (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 `@expand` 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 [B.4 STUB CONTENT](#) (55) 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 [B.4 STUB CONTENT \(55\)](#) 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 [B.4 STUB CONTENT \(55\)](#) for information on using this attribute.

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

Comment by robander

The definition above for `@keyref` should be synchronized with the definition in the linked section on keys.

Disposition: Unassigned

@keys

Specifies one or more names for a resource. See [B.4 STUB CONTENT \(55\)](#) 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 [B.4 STUB CONTENT \(55\)](#) 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](#).

Disposition: Unassigned

@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".

Disposition: Unassigned

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.

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.

Disposition: Unassigned

@name (data-element attributes)

Defines a unique name for the object.

Comment by robander

Do we need to specify the scope of "unique" here?

Disposition: Unassigned

@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](#) (55).

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 [B.4 STUB CONTENT](#) (55) 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.

Disposition: Unassigned

@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 [B.4 STUB CONTENT](#) (55) 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.

Disposition: Unassigned

@type (link-relationship attributes)

Describes the target of a reference. See [B.4 STUB CONTENT](#) (55) 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>, <line>, 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/".

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](#) (56)

[STUB CONTENT](#) (56)

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

C Formatting expectations

DITA is a standard that supports the creation of human-readable content. Accordingly, DITA defines fundamental document components. Since there is a reasonable expectation that such document components be formatted consistently, we suggest the following conventions.

Comment by Eliot Kimber

Do we plan to move these into the individual reference topics?

Kris Eberlein, 14 November 2022

No – All of this info was removed from the individual element-reference topics. I think what we need to revise is the short description – and perhaps we need to clarify the distinction between formatting and rendering. For me, formatting is up to the implementation and locale. Applying bold highlighting (and should this be emphasis?) is up for an implementation to decide what it means. Maybe company A uses bold, and company B colors the text purple.

And maybe we should change the title of the topic from "Formatting expectations" to "Formatting conventions" ...

Zoë 15 Nov

I agree with "conventions" over "expectations"

Kris Eberlein, 16 November 2022

I've changed the title to "Formatting conventions", and I've changed the short description to "Although how DITA elements are formatted is ultimately implementation-specific, certain conventions are common."

Disposition: Completed

Table 1: Formatting expectations for DITA elements

Element	Suggested formatting
<chdeschd>	Apply bold highlighting to the contents of the <chdeschd> element.
<choicetable>	Unless the @keycol attribute is set to "0", processors typically apply bold highlighting to the contents of the "Option" column.
<choptionhd>	Apply bold highlighting to the contents of the <choptionhd> element.
<codeblock>	Use a monospaced font for the contents of the <codeblock> element.
<codeph>	Use a monospaced font for the contents of the <codeph> element.
<menucascade>	Separate <uicontrol> elements with a character to represent the menu cascade.
<numcharref>	Surround the contents of the <numcharref> element with a leading ampersand (&) and a trailing semi-colon (;).

Element	Suggested formatting
<parameterentity> >	<p>Surround the contents of the <numcharref> element with a leading percentage sign (%) and a trailing semi-colon (;).</p> <div style="border: 1px solid black; padding: 5px;"> <p>Comment by Zoë Lawson on 15 Nov referencing the incorrect xml element. should be <parameterentity> Disposition: Unassigned</p> </div>
<screen>	Enclose the contents of the <screen> element with a box to suggest a computer display screen.
<shortcut>	Highlight the keyboard shortcut with underlining.
<syntaxdiagram>	Traditionally, the syntax diagram is formatted with "railroad tracks" that connect the units of the syntax together, but the presentation might differ depending on the output media.
<textentity>	Surround the contents of the <textentity> element with a leading ampersand (&) and a trailing semi-colon (;).
<var>	Apply italic highlighting to the contents of the <var> element.
<xmlatt>	Precede the contents of the <xmlatt> element with a commercial at symbol (@) or equals character (=) .
<xmlelement>	Surround the contents of the <xmlelement> element with leading (<) and trailing (>) angle brackets.
<>	

Index

A

attribute groups

- architectural attributes [25](#)
- common map attributes [25](#)
- complex table attributes [25](#)
- data-element attributes [25](#)
- date attributes [25](#)
- display attributes [25](#)
- ID and conref attributes [25](#)
- inclusion attributes [25](#)
- link relationship attributes [25](#)
- localization attributes [25](#)
- metadata attributes [25](#)
- simple table attributes [25](#)
- universal [38](#)
- universal attributes [25](#)

C

common attributes [43](#)

- `<conbody>` [9](#)
- `<conbodydiv>` [9](#)
- `<concept>` [8](#)
- concept elements [8](#)
 - `<conbody>` [9](#)
 - `<conbodydiv>` [9](#)
 - `<concept>` [8](#)

P

- `<propdesc>` [12](#)
- `<propdeschd>` [12](#)
- `<properties>` [13](#)
- `<property>` [15](#)
- property tables
 - descriptions [12](#)
 - headers
 - description columns [12](#)
 - type column [18](#)
 - value columns [19](#)
 - types [17](#)
 - values [18](#)
- `<prophead>` [16](#)
- `<proptype>` [17](#)
- `<proptypehd>` [18](#)
- `<propvalue>` [18](#)
- `<propvaluehd>` [19](#)

R

- `<refbody>` [19](#)

- `<refbodydiv>` [20](#)
- `<reference>` [20](#)
- reference elements
 - `<propdesc>` [12](#)
 - `<propdeschd>` [12](#)
 - `<properties>` [13](#)
 - `<property>` [15](#)
 - `<prophead>` [16](#)
 - `<proptype>` [17](#)
 - `<proptypehd>` [18](#)
 - `<propvalue>` [18](#)
 - `<propvaluehd>` [19](#)
 - `<refbody>` [19](#)
 - `<refbodydiv>` [20](#)
 - `<reference>` [20](#)
 - `<refsyn>` [22](#)
- `<refsyn>` [22](#)
- rendering expectations
 - `<properties>` [13](#)

T

- tables
 - properties [13](#)

U

- universal attribute group [38](#)