DITA Technical Committee

Editorial guidelines

Table of contents

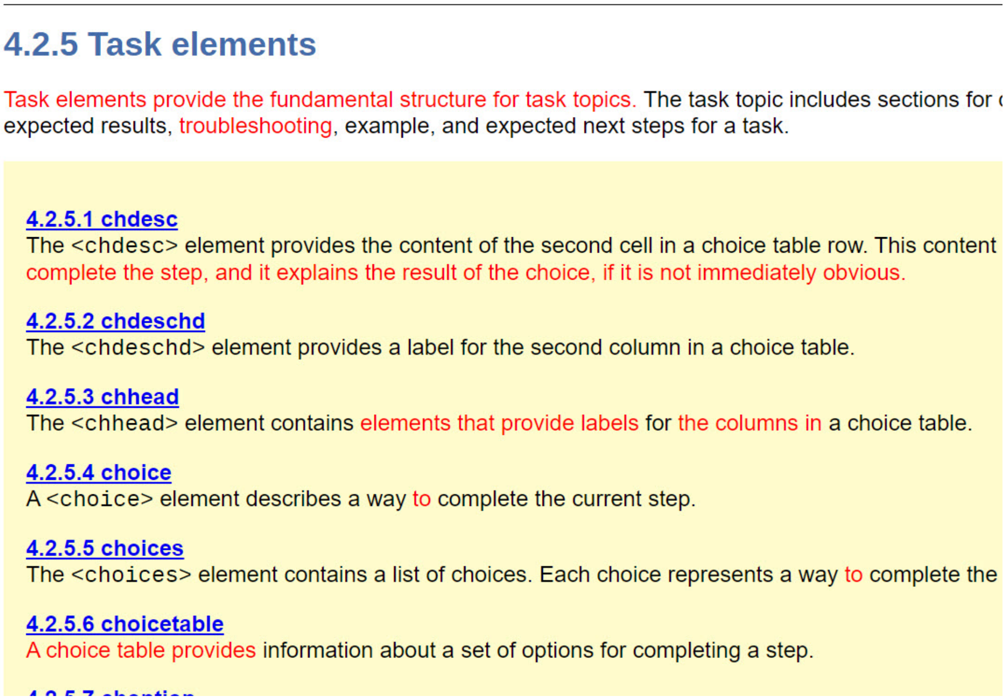
Element-reference topics

Each element-reference topic follows a set pattern. All topics have short descriptions, as well as "Attributes" and "Example" sections. These topics might also have "Usage information," "Rendering expectations," "Processing expectations," and "Specialization hierarchy" sections.

Short description

Use natural language for short descriptions, unless doing so would be incredibly awkward.

To check that the short descriptions for a cluster of topics are parallel, generate HTML5 output and look at the link previews. For example:



4.2.5 Task elements

Task elements provide the fundamental structure for task topics. The task topic includes sections for
expected results, troubleshooting, example, and expected next steps for a task.

4.2.5.1 chdesc
The <chdesc> element provides the content of the second cell in a choice table row. This content
complete the step, and it explains the result of the choice, if it is not immediately obvious.

4.2.5.2 chdeschd
The <chdeschd> element provides a label for the second column in a choice table.

4.2.5.3 chhead
The <chhead> element contains elements that provide labels for the columns in a choice table.

4.2.5.4 choice
A <choice> element describes a way to complete the current step.

4.2.5.5 choices
The <choices> element contains a list of choices. Each choice represents a way to complete the

4.2.5.6 choicetable
A choice table provides information about a set of options for completing a step.

4.2.5.7 choption

Examples of natural-language short descriptions

Consider the following short descriptions for the <steps> and <cmd> elements:

> Steps are a series of actions that people perform in a specific order and manner.
>
> A command specifies the action that people take to complete a step.

Examples of short description that do not use natural language

Consider the following short descriptions for the <chhead> and <choices> elements:

> The <chhead> element contains elements that provide labels for the columns in a choice table.
>
> The <choices> element contains a list of choices. Each choice represents a way to complete the current step.

Usage information

Use this section to provide additional information that is necessary for a DITA practitioner to understand.

Consider the following example for the <dita> element:

> The <dita> element cannot be specialized. It is provided as a container that can manage any sequence of any type of topic. Topic nesting rules can be configured in the document-type shell.

Convenience elements should have paragraphs that contain the following sort of information:

> The <linktitle> element is a convenience element. It is equivalent to a <titlealt> element with @title-role set to "linking".

Rendering expectations

This section explains rendering expectations for an element. In general, rendering expectations are about how the content will show up "on the glass."

Here are some examples from the <shortdesc> and <desc> topics:

> Processors *SHOULD* render the content of the <shortdesc> element as the initial paragraph of the topic.
>
> When used in conjunction with <fig> or <table> elements, processors *SHOULD* consider the content of <desc> elements to be part of the content flow.
>
> When used in conjunction with <xref> or <link> elements, processors often render the content of <desc> elements as hover help or other forms of link preview.

Note that the content in a "Rendering expectations" sections does not need to include normative RFC-2119 statements, although they often do.

Be sure to distinguish between rendering and formatting expectations; formatting expectations go in a non-normative appendix topic.

One way to distinguish between rendering and formatting is that rendering expectations are important for interoperability. For example, in certain cases, it is important that rendering applications are consistent in choosing what content to display: the <shortdesc> is rendered, and when an <object> cannot be displayed the <fallback> is rendered.

Formatting can vary without impacting the content itself, such as how indentation works for <dl> or how a <note> element is styled to stand out.

Processing expectations

Use this section to explain processing expectations for the element. In general, processing expectations help enforce consistency in how conforming DITA processors work with an element. Implementers need to pay special attention to any element that has this section.

Consider the following example from the <shortdesc> topic:

> When a <shortdesc> element occurs in a DITA map, it overrides the short description provided in the topic for the purpose of generating map-based link previews. It does not replace the <shortdesc> in the rendered topic itself. This means that generated map-based links to this topic will use the short description from the map for any link previews provided with the link, while the rendered topic continues to use the short description inside the topic.

If the processing expectations for the element are described in an architectural topic, link to that content. For example, the following paragraph is the sole content of "Processing expectations" topic in the `<keytext>` topic:

> See "Processing key references to generate text or link text".

Specialization hierarchy

(Specialized elements only) This section explains the specialization base for an element and lists the module in which the element is defined.

The wording for this section will vary, depending on the levels of specialization.

Level of specialization	Example
Typical	The `<codeph>` element is specialized from `<ph>`. It is defined in the programming domain module.
Domain that is specialized from another domain	The `<syntaxdiagram>` element is specialized from `<fig>`. It is defined in the syntax-diagram domain module, which is a specialization of the programming domain module.

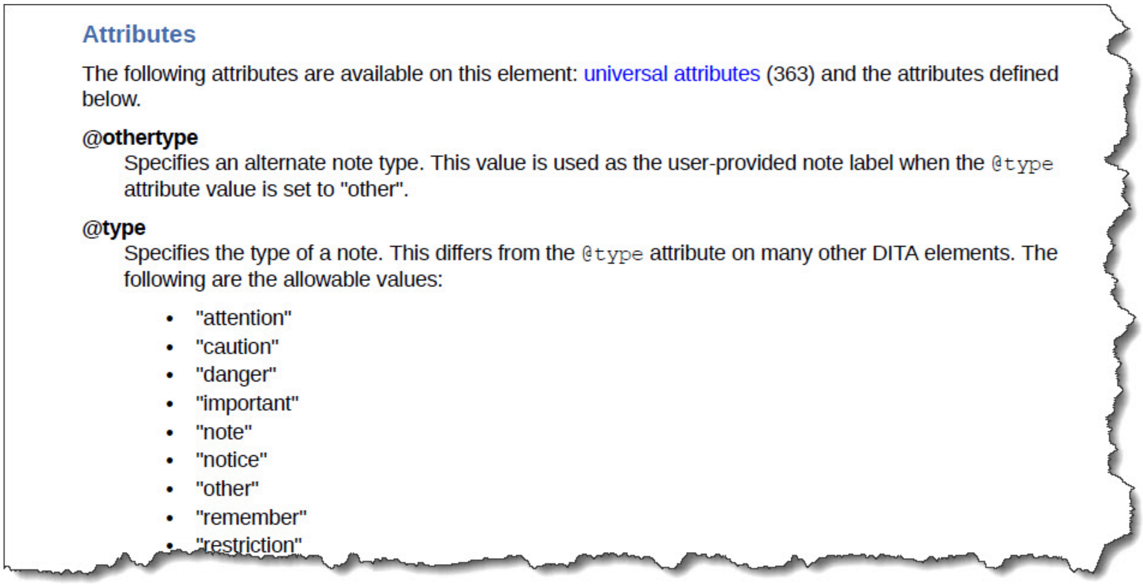
Attributes

This section provides information about the attributes that are available on an element.

Typically, the attributes are defined in a reuse topic and then conkeyrefed into the element reference topic.

The attributes are listed in alphabetical order, which attribute groups preceding individual attributes. Then, any attribute exceptions are listed. Finally, any attributes that are defined directly in the element-reference topic.

For example, consider the "Attributes" section from the `<note>` topic:



Attributes

The following attributes are available on this element: universal attributes (363) and the attributes defined below.

@othertype
Specifies an alternate note type. This value is used as the user-provided note label when the `@type` attribute value is set to "other".

@type
Specifies the type of a note. This differs from the `@type` attribute on many other DITA elements. The following are the allowable values:

- "attention"
- "caution"
- "danger"
- "important"
- "note"
- "notice"
- "other"
- "remember"
- "restriction"

Example

This section should be titled either "Example" or "Examples," depending on how many examples the section contains.

Consider the following guidelines for making the "Example" sections consistent across the specification

- Each code sample should be preceded by an explanatory paragraph that describes what the example illustrates. A good pattern came out of the LWDita work: "The following code sample shows how X does Y:" The intro paragraph should end with a colon, unless it is followed by another sentence (then use a period).
- The code sample should be indented two spaces. Use bold highlighting to call attention to the opening and closing tags.
- Code samples should be valid XML that can be copied-and-pasted into an editor (in the proper context). If ellipses are used to indicate missing information, surround them in an XML comment.
- (For sections with multiple examples) Begin with a paragraph that states "This section contains examples of how the `@elementName` element can be used. Surround each introductory paragraph and code sample with a `<figure>` element that has a descriptive title.
- (For elements that are part of a larger structure, such as a table or definition list entry) Instead of containing an example, the section can have a cross reference to another topic. Use the following type of markup:

  ```
  <p>See <xref keyref="elements-choicetable"><xmlelement>choicetable</xmlelement></
  xref>.</p>
  ```

- For examples where you want to show possible rendering, use a screen capture with good alternate text. Do not just include markup, leaving the rendering up to whatever stylesheets are used when output is generated.
- Do your best to replace "faked" examples with content that would conceivably be production content.