

Editorial guidelines

Table of contents

Rewriting short descriptions.....	3
Element-reference topics.....	4
Short description.....	4
Usage information.....	5
Rendering expectations.....	5
Processing expectations.....	6
Specialization hierarchy.....	6
Attributes.....	6
Example.....	7

Rewriting short descriptions

Sometimes it's difficult to decide whether a short description should use natural language. This topic provides some guidance.

Consider the short descriptions for the elements in the programming domain. When this domain went to review in January 2023, the short descriptions did not implement natural language. The following table contains commentary:

Element	Original short description	Comments	Revised short description
<apiname>	The <apiname> element identifies the name of an application programming interface (API), such as a Java class name or method name.	Poor candidate for natural language. A natural-language short description will tend to seem circular.	
<codeblock>	The <codeblock> element identifies lines of program code.	Do not use natural language. A <i>code block</i> has a specific meaning in programming that does not match how we use it in DITA.	
<codeph>	The <codeph> element identifies a code snippet.	The current short description is problematic, as the term <i>code snippet</i> has a specific meaning in programming that does not match how we use the term in DITA.	A code phrase is a small portion of source code, machine code, or text that is displayed in-line.
<coderef>	The <coderef> element references an external file that contains literal code.	This is iffy and needs to be approached carefully. The short description for <topicref> should be the model for how to recast this in natural language.	A code reference is the mechanism for referencing an external file that contains literal code.
<option>	The <option> element describes an option that can modify a command or a configuration.	This is iffy. The term <i>option</i> is too broad to use as a natural-language replacement, and <i>command or configuration option</i> seems cumbersome. Perhaps <i>command-line option</i> ? However, that might exclude ways in which current DITA users have used <option> to describe GUI parameters.	[?] A command-line option is a command used to pass a parameter to a program. These entries, also called command-line switches, can pass along cues for changing various settings or running commands in an interface.
<parname>	The <parname> element identifies the name of a parameter.	Poor candidate for natural language. A natural-language short description will tend to seem circular.	
<parml>	The <parml> element identifies a specialized definition list that is designed for documenting parameters.	Good candidate for natural language. It easily can move to the format "An item is a ..."	A parameter list is a specialized definition list that is designed for documenting parameters.

Element	Original short description	Comments	Revised short description
<plentry>	The <plentry> element contains one or more parameter terms and definitions.	Good candidate for natural language. It is a container element, and it easily can move to the format "An item contains ..."	A parameter-list entry contains one or more parameter terms and definitions.
<pd>	The <pd> element specifies a parameter definition within a parameter list entry.	This is iffy. We can model it after how we defined <dd>, but does that really add value?	[?] A parameter definition is a definition of a term that is defined in a parameter-list entry.
<pt>	The <pt> element specifies a parameter term within a parameter list entry.	This is iffy. We can model it after how we defined <dt>, but does that really add value?	[?] A parameter term is a term that is defined in a parameter-list entry.

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.

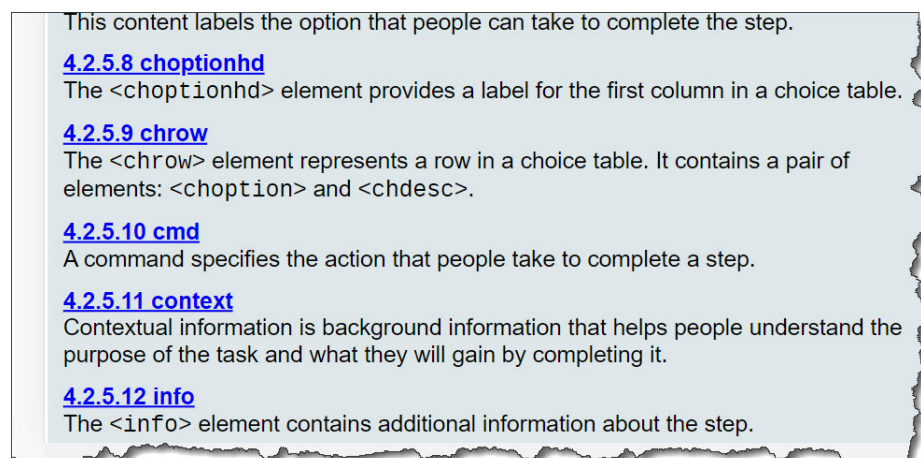
Short description

Use the short description to provide a description of "what something is". Use natural language, unless doing so would be incredibly awkward.

Tips for writing good short descriptions

For help on phrasing a natural language short description, perform a web search on the question "What is a *thing*". Often the search will provide a common definition very quickly.

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



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

Use this section to explain rendering expectations for an element. In general, rendering expectations are about how the content will show up "on the glass."

Here are some examples:

`<shortdesc>` topic

Processors *SHOULD* render the content of the `<shortdesc>` element as the initial paragraph of the topic.

`<desc>` 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 <code><codeph></code> element is specialized from <code><ph></code> . It is defined in the programming domain module.
Domain that is specialized from another domain	The <code><syntaxdiagram></code> element is specialized from <code><fig></code> . 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, with attribute groups preceding any 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

Markup

The <example> element should specify the @id and @otherprops attributes. Ensure that the following markup is used:

```
<example id="example" otherprops="examples">
```

This markup ensures the following:

- People can target the "Examples" section. The section might be a target of a cross reference by spec editors, or it might be a target of a conref push operation performed by companies that use the DITA spec source as part of their company-specific style guide.
- Implementers that want to publish the spec for their own purposes can choose to exclude non-normative content .

Introductory paragraphs and punctuation

Each code sample should be preceded by an explanatory paragraph that describes what the example illustrates. Use the following pattern: "The following code sample shows how X can be used to Y:" If the intro paragraph contains a single sentence, it should end with a colon. If the intro paragraph contains more than one sentence, end it with a period.

Examples:

The following code sample shows how the `<bodydiv>` element can be used to group a sequence of elements for reuse:

```
<body>
  <bodydiv id="mp-23475">
    <p>The maintenance pack includes the following items:</p>
    <ul>
      <li>Gloves</li>
      <li>Small screwdriver</li>
      <li>Part #23475</li>
    </ul>
  </bodydiv>
  <!-- ... -->
</body>
```

The following code sample shows how a short description can be used in a DITA map to provide information about a non-DITA resource. The content of the `<shortdesc>` element is used when a link preview to the Web site for the American Birding Association is generated.

```
<map>
  <title>Enjoying birds</title>
  <topicref href="birds-in-colorado.dita"/>
  <topicref href="bird-calling.dita"/>
  <topicref href="https://www.birding.example.com/" format="external" type="html">
    <topicmeta>
      <shortdesc>The American Birding Association is the only organization
        in North America that specifically caters to recreational birders.
        Its mission is to "inspire all people to enjoy and protect wild birds."
      </shortdesc>
    </topicmeta>
  </topicref>
</map>
```

Code sample indentation and highlighting

The code sample should be indented two spaces. Use bold highlighting to call attention to the opening and closing tags. If an example is displayed on a single line, highlight the element and its contents.

Examples:

The following code sample shows an `<abstract>` element that contains a short description, as well as additional phrase-level content:

```
<abstract>
  <shortdesc>
    Use the wonderful Widget to automatically vacuum your house.
  </shortdesc> It requires a 1800 lithium ion battery.
</abstract>
```

The following code sample shows how titles are used for both the topic and a figure within the topic:

```
<topic id="topicid">
  <title>Monitoring your heart rate</title>
  <body>
    <!-- ... -->
    <fig>
      <title>Adjusting your monitor</title>
      <p>If the monitor is not reporting, follow the directions
        in the video to adjust your equipment.</p>
    </fig>
  </body>
</topic>
```

Using ellipses to "snip" some of the code

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 example:


```

<abstract>
  <shortdesc>
    You have many options for arranging lodging in Brussels: hotels, bed and
    breakfasts, youth hostels, and flats. You can select from a wide price range.
  </shortdesc>
  <p>The following table explains the symbols that are used to indicate the price
  categories of the lodging options:</p>
  <simpletable>
    <!-- ... -->
  </simpletable>
</abstract>

```

Multiple examples

Begin with a paragraph that states "This section contains examples of how the <x> element can be used. Surround each introductory paragraph and code sample with a <figure> element that has a descriptive title. For example:

Examples

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

Figure 1: Short description in a topic

The following code sample shows how a <shortdesc> element can be used in a topic:

```

<topic id="intro-to-bird-calling">
  <title>Introduction to bird calling</title>
  <shortdesc>If you want to attract more birds to your Acme Bird Feeder, learn the art of
  bird calling. Bird calling is an efficient way to alert more birds to the presence of
  your bird feeder.
</shortdesc>
<body>
  <!-- ... -->
</body>
</topic>

```

Figure 2: Short description in a map

The following code sample shows how a short description can be used in a DITA map to provide information about a non-DITA resource. The content of the <shortdesc> element is used when a link preview to the Web site for the American Birding Association is generated.

```

<map>
  <title>Enjoying birds</title>
  <topicref href="birds-in-colorado.dita"/>
  <topicref href="bird-calling.dita"/>
  <topicref href="https://www.birding.example.com/" format="external" type="html">
    <topicmeta>
      <shortdesc>The American Birding Association is the only organization
      in North America that specifically caters to recreational birders.
      Its mission is to "inspire all people to enjoy and protect wild birds."
    </shortdesc>
    </topicmeta>
  </topicref>
</map>

```

Cross referencing to another topic

(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, where the key reference targets the "Example" section of the other target.

```

<p>See
  <xref keyref="elements-table/example"><xmlelement>table</xmlelement></xref>.
</p>

```

This markup will be rendered in the following way:

Example

See <table> (263).

If you need to link to a specific figure within an "Example" section, be sure to modify the target of the key reference so that you are linking to the specific figure.

Showing rendering of the code sample

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 the output is generated. (Remember that the DITA specification often is republished by other businesses and organizations who use their own stylesheets.)

Both the dita and dita-techcom repositories have an associated Word document that can be used to mock up rendered output. This document is named `screen-captures.docx`, and it located in the `specification/resources` directories.

Content of code samples

Do your best to replace "faked" examples with content that would conceivably be production content.