Concept Elements

This section outlines the main XML elements available in concept topics.

XML Elements

Element	Description
Concept	The <concept> element is the top-level element for a topic that answers the question "what is?" 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, but generally the structure of a concept is fairly simple. The model for a concept is as follows: `((title) then (titlealts) (optional) then (abstract or shortdesc) (optional) then (prolog) (optional) then (conbody) (optional) then (related-links) (optional) then (concept) (any number) )`
shortdesc	The short description <shortdesc> element occurs between the topic title and the topic body, as the initial paragraph-like content of a topic, or it can be embedded in an abstract element. The short description, which represents the purpose or theme of the topic, is also intended to be used as a link preview and for searching. Some guidelines to use short descriptions are included below: • Introduce the concept and provide a concise answer to the question "What is this?" and in some cases "Why do I care about this?" • If the concept is unfamiliar, you can start with a brief definition. • Describing the benefits of the tool, solution, part, or component. • Outlining a process. • Avoid using the short description to lead in or build up to a topic. The short description paragraph should contain the main point of the concept topic. • The concept short description should clearly apply to a concept. Avoid turning the concept topic into a task. • Do not simply repeat the title. • Do not use sentence fragments. Use complete sentences. • Avoid starting short descriptions with phrases such as "This topic describes . . . ." or "This topic is about . . . ."
conbody	The <conbody> element is the main body-level element for a concept. Like the <body> element of a general <topic>, <conbody> allows paragraphs, lists, and other elements as well as sections and examples. The content model is as follows: `((dl or parml or fig or syntaxdiagram or imagemap or image or lines or lq or note or hazardstatement or object or ol or p or pre or codeblock or msgblock or screen or simpletable or sl or table or ul or data or data-about or draft-comment or foreign or unknown or required-cleanup) (any number) then (section or example or conbodydiv) (any number))`

Element	Description

section	You can use <section> help users navigate and scan your content by organizing content into logical groups and providing headings, or section titles. Although the <title> element is optional for a <section> element, you should add titles in sections, which look like secondary headings in the output. The following shows the use of this element in XML:
	
	`<conbody>`
	`  <section>`
	`   <title>Purpose</title>`
	`   <p>This little command copies things.</p>`
	`  </section>`
	`</conbody>`

conbodydiv	The <conbodydiv> element is similar to the <<bodydiv>> element in that it provides an informal container for content that may be grouped within a concept. There are no additional semantics attached to the conbodydiv element; it is purely a grouping element provided to help organize content. Below is an example of its use:
	
	`<concept id="sample" xml:lang="en">`
	`<title>Conbodydiv example</title>`
	` <shortdesc>This concept is a sample of how to use`
	` conbodydiv.</shortdesc>`
	` <conbody>`
	`  <p>Introduce the example.</p>`
	`  <p>Next group some sections that may be reused`
	` elsewhere.</p>`
	`  <conbodydiv id="my_conbodydiv">`
	`   <section><title>First</title> ... </section>`
	`   <section><title>Second</title> ... </section>`
	`  </conbodydiv>`
	` </conbody>`
	`</concept>`

sl	The <sl> element contains a simple list of items of short, phrase-like content, such as a list of materials in a kit or package. Each bullet point is represented in a <sli>. On output, the list should have no bullets, on the assumption that each item is short enough to fit on one line, and needs no additional differentiation from its neighbors.
	
	Note: Use the Simple Lists (<sl> element) When You Don't Want to Show Bullets in the Output - If you use the simple list, limit the list items to short terms or phrases that are easy to scan without the visual aid of bullets.
	
	Below is an example of its use:
	
	`<section><title>Messages</title>`
	` <p>Messages from the ags_open module are identical with`
	` messages from:</p>`
	` <sl>`
	`  <sli>ags_read</sli>`
	`  <sli>ags_write</sli>`
	`  <sli>ags_close</sli>`
	` </sl>`
	`</section>`
	
	The following is what is outputted to the user:

Element	Description

ul	In an unordered list , the order of the list items is not significant. List items are typically styled on output with a "bullet" character, depending on nesting level. Each item in the list is represented in an .
	
	Note: Use Unordered Lists for Short, One-Part Items - If the content in your list item is more than one or two full sentences, consider presenting the information in another way, such as in a <section> or <dl> element.
	
	Below is an example of its use:
	
	`<ul>`
	`<li>This is an item in an unordered list.</li>`
	`<li>To separate it from other items in the list, the formatter puts a bullet beside it.</li>`
	`<li>The following paragraph, contained in the list item element, is part of the list item which contains it.`
	`<p>This is the contained paragraph.</p></li>`
	`<li>This is the last list item in our unordered list.</li>`
	`</ul>`

Element	Description

dl	A definition list <dl> is a list of terms and corresponding definitions. The term <dt> is usually flush left. The description or definition <dd> is usually either indented and on the next line, or on the same line to the right of the term. You can also provide an optional heading for the terms and definitions, using the <dlhead> element, which contains header elements for those columns. The default formatting for the <dlhead> generally looks like a table with a heading row, but this is also up to the rendering engine.
	
	Note: Use Definition Lists (<dl>element) for Two-Part List Items - Use definition lists (<dl> element) for lists of terms and their corresponding definitions or descriptions.
	
	Below is an example of its use:
	
	`<dl>`
	`<dlhead>`
	`<dthd>Image File View Selection</dthd>`
	`<ddhd>Resulting Information</ddhd>`
	`</dlhead>`
	`<dlentry>`
	`<dt>File Type</dt>`
	`<dd>Image's file extension</dd>`
	`</dlentry>`
	`<dlentry>`
	`<dt>Image Class</dt>`
	`<dd>Image is raster, vector, metafile or 3D</dd>`
	`</dlentry>`
	`<dlentry>`
	`<dt>Number of pages</dt>`
	`<dd>Number of pages in the image</dd>`
	`</dlentry>`
	`<dlentry>`
	`<dt>Fonts</dt>`
	`<dd>Names of the fonts contained within a vector image</dd>`
	`</dlentry>`
	`</dl>`
	
	The following is what is outputted to the user:

ol	An ordered list is a list of items sorted by sequence or order of importance.
	
	Note: Use Ordered Lists to Describe Workflows, Rankings, or High-Level Tasks That Aren’t Specific Steps - You can use ordered lists to describe processes or other information that entail a chronological progression. Ensure that an ordered list isn’t a user task disguised as a process. If you want users to follow a procedure, write a task topic.
	
	Below is an example of its use:
	
	`<ol>`
	`<li>Red</li>`
	`<li>Orange</li>`
	`<li>Yellow</li>`
	`<li>Green</li>`
	`<li>Blue</li>`
	`<li>Indigo</li>`
	`<li>Violet</li>`
	`</ol>`

Element	Description

parml	The parameter list <parml> element contains a list of terms and definitions that describes the parameters in an application programming interface. This is a special kind of definition list that is designed for documenting programming parameters. This element is part of the DITA programming domain, a special set of DITA elements designed to document programming tasks, concepts, and reference information. Below is an example of its use:
	
	`This code example is a basic method signature:`
	`<codeblock>returnType methodName(pList1, pList2) {</`
	`codeblock>`
	`where`
	`<parml>`
	`<plentry>`
	`<pt>pList1</pt>`
	`<pd>is the first variable declaration passed to`
	`methodName</pd>`
	`</plentry>`
	`<plentry>`
	`<pt>pList2</pt>`
	`<pd>is the second variable declaration passed to`
	`methodName</pd>`
	`</plentry>`
	`</parml>`
	
	The following is what is outputted to the user:

Element	Description
msgblock	The message block <msgblock> element contains a multi-line message or set of messages. The message block can contain multiple message numbers and message descriptions, each enclosed in a <msgnum> and <msgph> element. It can also contain the message content directly. Line breaks and spaces are preserved when the element is rendered. This element is part of the DITA software domain, a special set of DITA elements designed to document software tasks, concepts and reference information. Below is an example of its use inside a msgblock and p element: `<msgblock> <msgnum>I:0</msgnum> <msgph>informational: successful</msgph> </msgblock>  <p>A server log entry of <msgnum>I:0</msgnum> is equivalent to the text message, <msgph>informational: successful</msgph>. </p>` The following is what is outputted to the user: Other elements that can be used and included in msgbox include: • data • data-about • foreign • unknown • apiname • option • cmdname • varname • wintitle • term • abbreviated-form
term	Use the <term> element to mark up new terms that are introduced and defined in concept topics. Use the <term> element only for the first or most prominent occurrence of the term. By default, content in the <term> element is italicized in the output. The example below shows the XML for highlighting a new term: `<conbody> <section><title>Messages</title> It is probable that <q>temporary</q> or <q>new</q> stars, as these wonderful apparitions are called, really are <term>conflagrations</term>; not in the sense of a bonfire or a burning house or city, but in that of a sudden eruption of <i>inconceivable</i> heat and light, such as would result from the stripping off the shell of an encrusted sun or the crashing together of two mighty orbs flying through space with a hundred times the velocity of the swiftest cannon-shot. </section> </conbody>`

Element	Description
q and lq	The long quote <lq> element indicates content quoted from another source. Use the quote element <q> for short, inline quotations, and long quote <lq> for quotations that are too long for inline use, following normal guidelines for quoting other sources.

image	Include artwork or images in a DITA topic by using the <image> element. The <image> element has optional attributes that indicate whether the placement of the included graphic or artwork should be inline (like a button or icon) or on a separate line for a larger image. There are also optional attributes that indicate the size to which the included graphic or artwork should be scaled. An image element must specify an href attribute, a keyref attribute, or both. When both keyref and href are specified, the href is used as a fallback when the key reference cannot be resolved. The image addressed by the keyref or href is brought into the main flow of the content as rendered. To make the intent of the image more accessible for users using screen readers or text-only readers, authors should include a description of the image's content in the alt element. Below is an example of its use:
	`<fig>`
	`  <image keyref="Figure48" align="center" alt="Figure 48"`
	`scalefit="yes"></image>`
	`</fig>`

Others	This list is too exhaustive for me to go through each element. Please see the full conbody specification from OASIS.