Review S: Appendixes

Table of contents

A Acknowledgments.. 3
B Coding practices for DITA grammar files..4
B.1 File naming conventions..4
C Constraint modules.. 6
D Expansion modules..7
E Element-by-element recommendations for translators..8
F Formatting expectations.. 23
G OASIS grammar files.. 25
G.1 File names in the base DITA edition.. 25
G.2 Globally-unique identifiers in the base DITA edition.. 27
G.3 Domains provided in the base DITA edition..27
G.4 Document-type shells provided in the base DITA edition.. 28

A Acknowledgments

(Non-normative) Many current and past members of the OASIS DITA Technical Committee participated in the creation of this specification and are gratefully acknowledged.

Robert Anderson, Oracle
Deb Bissantz, Vasont Systems
Bill Burns, HP Inc.
Carsten Brennecke, SAP SE
Stan Doherty, Individual member
Kristen James Eberlein, Eberlein Consulting LLC
Carlos Evia, Virginia Tech
Nancy Harrison, Individual member
Alan Houser, Individual member
Scott Hudson, ServiceNow
Gershon Joseph, Precision Content Authoring Solutions Inc.
Eliot Kimber, ServiceNow
Zöe Lawson, Synopsys, Inc.
Tom Magliery, JustSystems
Chris Nitchie, Individual member
Keith Schengili-Roberts, Individual member
Eric Sirois, IXIASOFT
Dawn Stevens, Comtech Services, Inc.
Bob Thomas, Individual member
Frank Wegmann, Individual member

In addition, the OASIS DITA Technical Committee also would like to recognize the following people for their insights and support:

Silke Achterfeld
Robert Johnson
Jarno Elovirta

The DITA Technical Committee used the following applications to work with the DITA source:

- Antenna House Formatter
- Congility Content Server
- DITA Open Toolkit
- Oxygen Content Fusion
- Oxygen XML Editor
- XMetaL Author Enterprise

We are grateful to Antenna House for providing licenses for Antenna House Formatter, Mekon for providing an instance of Congility Content Server, and Syncro Soft for the use of Oxygen Content Fusion.

B Coding practices for DITA grammar files

This section contains information about creating modular DTD- or RELAX NG-based grammar files. It explains how document-type shells, specialization modules, and element-configuration modules (constraint and expansion) are organized.

B.1 File naming conventions

The OASIS DITA Technical Committee uses certain conventions for the names of XML grammar files. We suggest using these conventions as a way to facilitate the interchange of grammar files.

Globally unique identifiers

Vocabulary modules that are intended for use outside of a narrowly-restricted context should have one or more associated, globally-unique names by which the modules can be referenced without regard to their local storage location. The globally-unique names can be public identifiers, URNs, or absolute URLs.

Document-type shells

Document-type shells should be given a name that distinguishes their name, owner, or purpose, for example, `acme-concept.dtd`. The document-type shells that are provided by the DITA Technical Committee typically use the root element of the primary specialization as the basis for the file name. If necessary, a qualifier such as "base" is prepended to the name of the root element.

Comment by Eliot Kimber
I don't object to this convention but I've never followed it in my own work because the practice of organizing modules into OT plug-ins is sufficiently distinguishing. Within the context of such a plug-in, it's useful to retain the original unqualified file names for shells because adding more distinction doesn't help (or at least it doesn't help me).

In a typical environment you would normally only have the TC-defined shells and your local shells, one for each topic or map type you use, so the chance for confusion by having more than two "topic.dita" files is minimal.

By not following this convention, the only thing you have to modify when setting up local shells is the public ID in catalogs and comments--you don't have to also modify file names.

Kris Eberlein, 08 October 2022

I certainly don't disagree with you here. The main times that I find this convention useful is when both an OASIS document-type shell AND a company-specific document-type shell for the same information type are in play – for example, a company uses both their own shell for topic and the OASIS document-type shells for topic.

Maybe we should remove this guidance? Note that we do name the document-type shell for topic in the base differently that the document-type shell for topic in technical content ...

Disposition: Unassigned

Module names

For structural modules, the module name should be the element type name of the top-level topic or map type that is defined by the module, such as "topic" or "map".

For element- or attribute-domain modules, the module name should be a name that reflects the subject domain to which the domain applies, such as "highlight" or "software". Domain module names should be sufficiently unique that they are unlikely to conflict with any other domains.

In addition, each element- or attribute-domain module has a short name that is used to construct entity names that are used in associated declarations. Modules can also have abbreviated names that further shorten the short name, for example "hi-d" for the "highlight" domain, where "highlight" is the short name and "hi-d" is the abbreviated name.

C Constraint modules

This section of the specification contains examples of constraint modules implemented using both DTD and RNG.

Related concepts

Constraints

D Expansion modules

This section of the specification contains examples of expansion modules implemented using both DTD and RNG.

Related concepts

Expansion modules

E Element-by-element recommendations for translators

This topic contains a list of all elements that are available in the base DITAedition. It includes recommendations on how to present the element type to translators, whether the element contents are likely to be suitable for translation, and whether the element has attributes whose values are likely to be suitable for translation. Examples of content that is not suitable for translation include code fragments and mailing addresses.

Comment by Dawn Stevens on 04 October 2022

[First sentence, "DITAedition"] Add space between words

Kris Eberlein, 04 October 2022

Done

Disposition: Completed

Comment by Dawn Stevens on 04 October 2022

[Second sentence] It's a pet peeve of mine to use "whose" for objects that are not people. I would rewrite "has attributes with values that are likely"

Kris Eberlein, 04 October 2022

Done

Disposition: Completed

Comment by Dawn Stevens on 04 October 2022

[Second sentence"] In the many tables that don't have attributes suitable for translation, suggest that the column be removed and maybe a note added to the notes section that says if the column is missing that there are no translatable attributes.

Disposition: Unassigned

Since the distinction between block and inline elements is ultimately controlled by the container of the element and the processing associated with it, the same element might be a block in one context and an inline element in another. Specializing document types might vary this behavior according to the needs of the document type being created, and the distinctions given below are provided only as a guide to known behavior with the base DITA document types.

Comment by Dawn Stevens on 04 October 2022

This information feels to me to be too separated from the relevant notes in the next section that explains the block/inline columns. I feel like it should be a part of the notes section at least.

WEK: I think I agree. Could be added to the list of notes in the next section.

Disposition: Unassigned

Comment by Eliot Kimber
The discussion uses the term "subflow", which is used in a way that suggests it's a term of art that localizers would understand but I don't see a definition of here. Do we need to define *subflow* in this context?
Disposition: Unassigned

Notes on the tables below

- For specializations, the second column gives the ancestor element, and the third column gives a quick yes/no guide to indicate whether all behavior is inherited. If something is not inherited, the change will appear in bold.

  Comment by Eliot Kimber
  The source of the inheritance is not clear: is it the specialization ancestor or the XML containing ancestor?

  But beyond that, it's not clear what this column is telling me that is actionable or interesting: for the purposes of translation, why would it matter--ultimately the only thing that matters as far as I can tell is whether an element A) contains translatable text and B) is a block, and inline, or a subflow.

  In addition, it would be clearer to have a per-column description that references the column heading text: It took me a bit of hunting and re-reading to realize this bullet provides the definition of the "Inherits everything" column--searching on that text did not get me here.

  Disposition: Unassigned
- For any specialization not listed below, the suggested default is to fall back to the closest listed ancestor.
- The block/inline presentation column indicates whether the element is formatted as a single block.

  Comment by Eliot Kimber
  See my comments below.

  This statement is not really accurate because the presentation column really indicates if an element is an inline, block, or metadata element that is not rendered by default, and if rendered, not rendered in isolation (for the most part).

  Disposition: Unassigned
- The block/inline translation column indicates whether the element represents a complete translatable segment. For example, the element `<cmd>` is presented inline with other elements, but represents a complete translation segment.

  Comment by Dawn Stevens on 04 October 2022

  I'm going to say that I don't understand the distinction between the block/inline columns. In all cases in this document they are identical, except that the metadata elements have a parenthetical (metadata) in the presentation column.

  That said, the example of <cmd> in this bullet highlights that <cmd> is actually not listed in this element-by-element reference; I assume that is because it is not part of base dita, but technical communication. So it is not a valid example for this context, which brings me back to my original comment. for base dita, perhaps, these columns have no distinction?

WEK:

I think the block/inline (presentation) column should be "block/inline/metadata/subflow" as these are the presentation distinctions, which are different from the block/inline (translation) column, which I understand to just indicate whether you translate the entire element content as a single segment (block) or as part of a larger segment (inline).

Disposition: Unassigned

- Items marked as block*** are blocks on their own, but might appear in the middle of a segment. They should not break the flow of the current segment. These are considered "subflow" elements for translation.

  Comment by Eliot Kimber
  See my comment above about the definition of "sublow".

  I think it would be clearer to simply define subflow and then use that value instead of "block***" in the presentation column, i.e.:

  Some elements are "subflow" elements, which act as blocks for translation purposes but are presented out of line with their containing element and therefore do not break the flow of their containing element, e.g., footnotes (`<fn>`).

  Disposition: Unassigned

  We recommend that, when possible, these elements should only be placed at sentence boundaries to aid in translation.
- For all elements, the `@translate` attribute will override the suggested default translation setting. So, a translation setting of "yes" or "no" in the table below does not guarantee that an element will always, or never, be translated.

  Comment by Eliot Kimber
  c/will override/overrides/ and add ", if specified:

  …, the `@translate`attributes overrides the suggested default translation setting, if specified.

  Disposition: Unassigned
- If an element has translatable attributes, they are listed in the last column.

  Comment by Eliot Kimber
  To Dawn's point about translatable attributes: Almost no elements have translatable attributes and only a few attributes are appropriately translatable.

[1] This element is considered a "subflow" element for translation. If it is located in the middle of a translation segment, it should not be translated as part of that segment.

Comment by Eliot Kimber
The antecedent of "This element" is ambiguous--is it referring to the footnote element itself or to the elements to which "block***" applies? In any case, this footnote is redundant with the preceding sentence. The example within the footnote is probably better given in the main bullet.
Disposition: Unassigned

For example, `<indexterm>`, `<fn>`, and `<draft-comment>` might divide a sentence in two, but should be treated as blocks, and should not interrupt the sentence.

Probably more effective to have a separate section specifically for translatable attributes and omit the 99.9% empty column from the tables.

Disposition: Unassigned

- The <keyword> element (as well as specializations of <keyword>) is an inline, phrase-like element when it appears in the body of a document. It can also appear in the <keywords> element in <topicmeta> (for maps) or in the <prolog> (for topic). When it appears in the <keywords> element, each <keyword> represents an individual segment, and is not part of a larger segment; in that location, <keyword> can be considered a "subflow" element.

Topic elements

Comment by Dawn Stevens on 04 October 2022

In some cases in the tables "no" is plain text, sometimes bolded, sometimes capped and bolded. Are these formatting distinctions supposed to mean something?

Disposition: Unassigned

Comment by Dawn Stevens on 04 October 2022

I guess I really don't understand the table, why do elements like <ul>, <ol>, <vrmlist> (and many many others) which do not contain content show they have translatable content? Technically it's <li> in the first two that have translatable content. In the later, <vrm> doesn't have translatable content either, just attributes which really aren't meant to be translatable. Why are elements that should contain numbers (like copyryear) or names (author) translatable? I started marking and questioning, but removed my comments, deciding I didn't understand what the table was truly deciding to do.

WEK:

I understand the "Translatable content" column to mean "Does (or can) this element contain elements that require translation?" rather than "Does this element directly contain text that requires translation".

So I guess the question is: is it necessary to identify elements that contain elements that contain translatable text? I assume the current design is intended to provide a complete element-to-rule mapping.

Maybe a way to better clarify the distinction would be to add the category "container" to the block/inline (translation) list, to identify elements that only contain other elements that require translation, i.e., <ul>, <dlentry>, <body>, etc.

Disposition: Unassigned

Comment by Eliot Kimber
Looking at the entry for <alt>, it seems like the block/inline (presentation) column could be block/inline/subflow and then just use "subflow" rather than the xref to the sentence that discusses subflow. Or maybe block/inline/subflow/metadata?

[2] The block vs. inline designation for the <foreign> element is likely to change for some specializations.

[3] The <desc>, <object>, and <image> elements inside <foreign> should still be translatable; they provide an alternative display if the foreign content cannot be processed.

Disposition: Unassigned

Element name	Specialized from	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<abstract>`	N/A	block	block	yes	
`<alt>`	N/A	block***[1]	block	yes	
`<audience>`	N/A	block (metadata)	block	yes	
`<audio>`	N/A	block	block	yes	
`<author>`	N/A	block (metadata)	block	yes	
`<body>`	N/A	block	block	yes	
`<bodydiv>`	N/A	block	block	yes	
`<brand>`	N/A	block (metadata) **Comment by Eliot Kimber** As this is the "presentation" column I think that just "metadata" would be clearer give that the next column indicates the "block/inline" for translation purposes. Thus "metadata" means "not presented by default". Having "block" here just seems to confuse things as these elements are not otherwise "block" elements for presentation purposes. **Disposition: Unassigned**	block	yes	
`<category>`	N/A	block (metadata)	block	yes	
`<cite>`	N/A	inline	inline	yes	

Element name	Specialized from	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<colspec>`	N/A	n/a	n/a	n/a	
`<component>`	N/A	block (metadata)	block	yes	
`<copyrholder>`	N/A	block (metadata)	block	yes	
`<copyright>`	N/A	block (metadata)	block	yes	
`<copyryear>`	N/A	block (metadata)	block	yes	
`<created>`	N/A	block (metadata)	block	yes	
`<critdates>`	N/A	block (metadata)	block	yes	
`<data>`	N/A	N/A (metadata)	block	no (likely to change for some specializations)	
`<dd>`	N/A	block	block	yes	
`<ddhd>`	N/A	block	block	yes	
`<desc>`	N/A	block	block	yes	
`<div>`	N/A	block	block	yes	
`<dl>`	N/A	block	block	yes	
`<dlentry>`	N/A	block	block	yes	
`<dlhead>`	N/A	block	block	yes	
`<draft-comment>`	N/A	block***[1]	block	**no**	
`<dt>`	N/A	block	block	yes	
`<dthd>`	N/A	block	block	yes	
`<entry>`	N/A	block	block	yes	
`<example>`	N/A	block	block	yes	
`<fallback>`	N/A	block	block	yes	
`<featnum>`	N/A	block (metadata)	block	yes	
`<fig>`	N/A	block	block	yes	
`<figgroup>`	N/A	block	block	yes	
`<fn>`	N/A	block***[1]	block	yes	
`<foreign>` [5]	N/A	block[2]	block[2]	no[3]	

[4] The `<desc>`, `<object>`, and `<image>` elements inside `<foreign>` should still be translatable; they provide an alternative display if the foreign content cannot be processed.

[5] The block vs. inline designation for the `<foreign>` element is likely to change for some specializations.

Element name	Specialized from	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
				Comment by Eliot Kimber Instead of "no" why not "May contain DITA elements in addition to non-DITA elements" **Disposition: Unassigned**	
`<image>`	N/A	block when `@placement`= break, otherwise inline	block when `@placement`= break, otherwise inline **Comment by Eliot Kimber** `<image>` is not a block for translation purposes, it is a container--only the `<alt>` element within `<image>` contains translatable text. Localization of the referenced image would be separate. **Disposition: Unassigned**	yes	
`<include>`	N/A	inline	inline	yes	
`<index-see>`	N/A	block***[1]	block	yes	yes **Comment by dstevens73** In all other cases, the translatable attribute is

Element name	Specialized from	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
					listed, except here and on the next row. **Disposition: Unassigned**
<index-see-also>	N/A	block***1	block	yes	yes
<indexterm>	N/A	block***1	block	yes	
<keytext>	N/A	block	block	yes	
<keyword>	N/A	inline	inline (except when within <keywords> – see note above the table)	yes	
<keywords>	N/A	block	block	yes	
<li>	N/A	block	block	yes	
<lines>	N/A	block	block	yes	
<link>	N/A	block	block	yes	
<linkinfo>	N/A	block	block	yes	
<linklist>	N/A	block	block	yes	
<linkpool>	N/A	block	block	yes	
<linktext>	N/A	block	block	yes	
<lq>	N/A	block	block	yes	@reftitle
<media-source>	N/A	block	block	n/a **Comment by Eliot Kimber** How is n/a different from "no"? **Disposition: Unassigned**	
<media-track>	N/A	block	block	n/a	
<metadata>	N/A	block (metadata)	block	yes	
<no-topic-nesting>	N/A	n/a	n/a	n/a	
<note>	N/A	block	block	yes	@othertype **Comment by Eliot Kimber** How is @othertypetranslatable? It would normally be used

Element name	Specialized from	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
					as a keyword to the rendering to then generate the appropriate label, which would be localized in the rendition processing. Expecting authors to provide the presentation label in @othertype would be very bad practice. **Disposition: Unassigned**
<object>	N/A	block	block	yes	
<ol>	N/A	block	block	yes	
<othermeta>	N/A	block (metadata)	block	yes	@content
<p>	N/A	block	block	yes	
<param>	N/A	block	block	n/a	
<permissions>	N/A	block (metadata)	block	yes	
<ph>	N/A	inline	inline	yes	
<platform>	N/A	block (metadata)	block	yes	
<pre>	N/A	block	block	yes	
<prodinfo>	N/A	block (metadata)	block	yes	
<prodname>	N/A	block (metadata)	block	yes	
<prognum>	N/A	block (metadata)	block	yes	
<prolog>	N/A	block (metadata)	block	yes	
<publisher>	N/A	block (metadata)	block	yes	
<q>	N/A	inline	inline	yes	
<related-links>	N/A	block	block	yes	
<required-cleanup>	N/A	block***[1]	block	**no**	
<resourceid>	N/A	block (metadata)	block	yes	
<revised>	N/A	block (metadata)	block	yes	
<row>	N/A	block	block	yes	
<section>	N/A	block	block	yes	
<series>	N/A	block (metadata)	block	yes	
<shortdesc>	N/A	block	block	yes	
<simpletable>	N/A	block	block	yes	

Element name	Specialized from	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<sl>`	N/A	block	block	yes	
`<sli>`	N/A	block	block	yes	
`<source>`	N/A	block (metadata)	block	yes	
`<state>`	N/A	inline	inline	yes	`@value`
`<stentry>`	N/A	block	block	yes	
`<sthead>`	N/A	block	block	yes	
`<strow>`	N/A	block	block	yes	
`<table>`	N/A	block	block	yes	
`<tbody>`	N/A	block	block	yes	
`<term>`	N/A	inline	inline	yes	
`<text>`	N/A	inline	inline	yes	
`<tgroup>`	N/A	block	block	yes	
`<thead>`	N/A	block	block	yes	
`<title>`	N/A	block	block	yes	
`<titlealt>`	N/A	block	block	yes	
`<tm>`	N/A	inline	inline	yes	
`<topic>`	N/A	block	block	yes	
`<ul>`	N/A	block	block	yes	
`<unknown>`	N/A	block	block	no	
`<video>`	N/A	block	block	yes	
`<vrm>`	N/A	block (metadata)	block	yes	
`<vrmlist>`	N/A	block (metadata)	block	yes	
`<xref>`	N/A	inline	inline	yes	

Map elements

Element name	Specialized from	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<map>`	N/A	block **Comment by Eliot Kimber** To be annoyingly pedantic, no map element is directly presented in the way that topic elements are so these should all be "n/a".	block	yes	

Element name	Specialized from	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
		Maps are 100% metadata that is presented only indirectly through some presentation process. **Disposition: Unassigned**			
`<navref>`	N/A	n/a	n/a	n/a	
`<relcell>`	N/A	block	block	yes	
`<relcolspec>`	N/A	block	block	yes	
`<relheader>`	N/A	block	block	yes	
`<relrow>`	N/A	block	block	yes	
`<reltable>`	N/A	block	block	yes	
`<shortdesc>`	N/A	block	block	yes	
`<topicmeta>`	N/A	block	block	yes	
`<topicref>`	N/A	block	block	yes	
`<ux-window>`	N/A	N/A (empty)	N/A (empty)	no	

Alternative Title Elements

Element name	Specialized from	Inherits everything from ancestor?	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<linktitle>`	`<titlealt>`	yes	N/A (metadata)	block	yes	
`<navtitle>`	`<titlealt>`	yes	N/A (metadata)	block	yes	
`<searchtitle>`	`<titlealt>`	yes	N/A (metadata)	block	yes	
`<subtitle>`	`<titlealt>`	yes	block	block	yes	
`<titlehint>`	`<titlealt>`	yes	N/A (metadata)	block	yes	

Emphasis domain elements (emphasis-d)

Element name	Specialized from	Inherits everything from ancestor?	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<em>`	`<ph>`	yes	inline	inline	yes	
`<strong>`		yes	inline	inline	yes	

Element name	Specialized from	Inherits everything from ancestor?	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
	Comment by Eliot Kimber Use <xmlelement>ph</xmlelement> **Disposition: Unassigned**					

Hazard statement domain (hazard-d elements)

Element name	Specialized from	Inherits everything from ancestor?	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<consequence>`	`<li>`	yes	block	block	yes	
`<hazardstatement>`	`<note>`	yes	block	block	yes	`@othertype` **Comment by Eliot Kimber** Same comment as above: `@othertype` values should be keywords localized by the renderer and not authored with the expectation of being localized. **Disposition: Unassigned**
`<hazardsymbol>`	`<image>`	yes	block when `@placement=` break, otherwise inline	block when `@placement=` break, otherwise inline **Comment by Eliot Kimber** See comment on `<image>`	yes	

Element name	Specialized from	Inherits everything from ancestor?	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
				Disposition: Unassigned		
`<howtoavoid>`	`<li>`	yes	block	block	yes	
`<messagepanel>`	`<ul>`	yes	block	block	yes	
`<typeofhazard>`	`<li>`	yes	block	block	yes	

Highlight domain elements (hi-d)

Element name	Specialized from	Inherits everything from ancestor?	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<b>`	`<ph>`	yes	inline	inline	yes	
`<line-through>`	`<ph>`	yes	inline	inline	yes	
`<i>`	`<ph>`	yes	inline	inline	yes	
`<overline>`	`<ph>`	yes	inline	inline	yes	
`<sub>`	`<ph>`	yes	inline	inline	yes	
`<sup>`	`<ph>`	yes	inline	inline	yes	
`<tt>`	`<ph>`	yes	inline	inline	yes	
`<u>`	`<ph>`	yes	inline	inline	yes	

Utilities domain elements

Element name	Specialized from	Inherits everything from ancestor?	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<area>`	`<figgroup>`	yes	block	block	yes	
`<coords>`	`<ph>`	**NO** **Comment by Eliot Kimber** All-caps "NO". Should be "no". Also entries below for shape and short-as **Disposition: Unassigned**	inline	inline	**no** **Comment by Eliot Kimber** To Dawn's comment about why some of these are bold--this is bold because it's different from its	

Element name	Specialized from	Inherits everything from ancestor?	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
					specialization ancestor. However, I'm not sure it's necessary to highlight it. **Disposition: Unassigned**	
`<imagemap>`	`<fig>`	yes	block	block	yes (can contain translatable alternate text)	
`<shape>`	`<keyword>`	**NO**	inline	inline	**no**	
`<sort-as>`	`<data>`	**NO**	block***[1]	block	yes	

DITAVALref domain elements

Element name	Specialized from	Inherits everything from ancestor?	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<ditavalmeta>`	`<topicmeta>`	yes	block	block	yes **Comment by Eliot Kimber** I guess this should be "yes" because it can contain "navtitle", which I suppose could be usefully translated if present. Same reasoning for `<ditavalref>` as it can contain `<ditavalmeta>` **Disposition: Unassigned**	
`<ditavalref>`	`<topicref>`	yes	block	block	yes	
`<dvrKeyscopePrefix>`	`<data>`	yes	N/A (metadata)	block	no	
`<dvrKeyscopeSuffix>`	`<data>`	yes	N/A (metadata)	block	no	
`<dvrResourcePrefix>`	`<data>`	yes	N/A (metadata)	block	no	

Element name	Specialized from	Inherits everything from ancestor?	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<dvrResourceSuffix>`	`<data>`	yes	N/A (metadata)	block	no	

Map group domain elements (mapgroup-d)

Element name	Specialized from	Inherits everything from ancestor?	Block/Inline (presentation)	Block/Inline (translation)	Translatable content?	Translatable attributes?
`<keydef>`	`<topicref>`	yes	block	block	yes	
`<mapref>`	`<topicref>`	yes	block	block	yes	
`<mapresources>`	`<topicref>`	yes	block	block	yes	
`<topicgroup>`	`<topicref>`	yes	block	block	yes	
`<topichead>`	`<topicref>`	yes	block	block	yes	

DITAVAL elements

The DITAVAL elements are not specialized, and are not rendered on their own, so related columns are dropped from this table. There are no translatable attributes in the DITAVAL element set.

The only element that directly contains text for translation is `<alt-text>`.

Element name	Block/Inline (translation)	Translatable content?
`<alt-text>`	block	yes
`<endflag>`	block	yes (inside nested elements)
`<prop>`	block	yes (inside nested elements)
`<revprop>`	block	yes (inside nested elements)
`<startflag>`	block	yes (inside nested elements)
`<style-conflict>`	block	N/A *(empty element)*
`<val>`	block	yes (inside nested elements)

F 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 rendered consistently, we suggest the following formatting conventions.

Comment by Eliot Kimber
I tend to prefer "as" instead of "since" and tried to find support for that and in fact found the opposite: reference to at least one authority that prefers "since" over "as". As I can't find an authority to support my preference for "as", I will not suggest "correcting" "since" to "as" here and elsewhere.
Disposition: Unassigned

Table 1: Formatting expectations for DITA elements

Element	Suggested formatting
`<b>`	Apply bold highlighting to the contents of the `<b>` element.
`<cite>`	Set citations apart from the surrounding text by a form of highlighting, for example, italics.
`<dd>`	See `<dl>`.
`<dl>`	Apply the following conventions: • The term (`<dt>`) is against the starting margin of the page or column. • The description or definition (`<dd>`) is either indented and on the next line or on the same line after the term. • The `<dlhead>` looks like a table heading row.
`<dlhead>`	See `<dl>`.
`<dt>`	See `<dl>`.
`<em>`	For Western languages, apply italic highlighting to the contents of the `<em>` element.
`<i>`	For Western languages, apply italic highlighting to the contents of the `<i>` element.
`<li>`	Apply the following conventions: • In ordered lists, list items are indicated by numbers or alphabetical characters. • In unordered lists, list items are indicated by bullets or dashes.
`<lines>`	Render the contents of `<lines>` elements in a non-monospaced font. **Comment by Eliot Kimber** Should we mention preservation of line breaks and white space? It's implicit in the `@xml:space` value for the element but you'd only know that if you looked at the grammar. Kris Eberlein, 08 October 2022 I think that is information that should be in the "Rendering expectations" section for the lines topic. **Disposition: Unassigned**
`<line-through>`	Render the contents of the `<line-through>` element with a line struck through.

Element	Suggested formatting
<lq>	Render the contents of the <lq> element as an indented block.
<note>	Render a label for notes. The content of the label depends on the values of the @type attribute. A note typically is formatted in a way that stands out from the surrounding content.
<ol>	See <li>.
<overline>	Render a line above the contents of the <overline> element.
<pre>	Render the content of a <pre> element in a monospaced font. **Comment by Eliot Kimber** Line break and whitespace preservation? Kris Eberlein, 08 October 2022 I think that is information that should be in the "Rendering expectations" section for the pre topic. **Disposition: Unassigned**
<sl>	See <sli>.
<sli>	Apply the following conventions: • The content of each simple list item is placed on a separate line. • The lines are not distinguished by numbers, bullets, or other icons.
<strong>	Apply bold highlighting to the contents of the <strong> element.
<sub>	Render the contents of the <sub> element lower in relationship to the surrounding text and in a smaller font.
<sup>	Render the contents of the <sup> element higher in relationship to the surrounding text and in a smaller font.
<tt>	Render the contents of the <tt> element in a monospaced font.
<u>	Apply underlining to the contents of the <u> element.

G OASIS grammar files

This section provides information about the grammar files that are provided in the DITA base edition.

G.1 File names in the base DITA edition

The OASIS DITA Technical Committee uses certain conventions for the names of XML grammar files. We suggest using these conventions as a way to facilitate the interchange of grammar files.

DTD-based specialization modules

The DITA Technical Committee uses certain file-naming conventions for DTD-based specialization modules. While the grammar files shipped with DITA 2.0 do not include domain constraint or expansion modules, we suggest conventions for those modules also.

Module type	File name	Example
Structural	*moduleName*.mod	topic.mod
Element domain	*domainName*Domain.*ext*	highlightDomain.ent highlightDomain.mod
Attribute domain	*attriName*AttDomain.ent	deliveryTargetAttDomain.ent
Constraint	*qualifierTarget*Constraint.mod	strictTaskbodyConstraint.mod acmeHighlightDomainConstraint.mod
Expansion		acme-SectionExpansion.mod acme-CellPurposeAttExpansion.ent acme-otherpropsAttExpansion.mod example-dlentryModeAttExpansion.ent

Comment by Kristen J Eberlein on 19 September 2022

The names of the expansion modules listed in the "Example" column are taken from the example topics. They do not follow a consistent pattern. I suspect that the same is true for file names used in the constraint example topics.

Disposition: Unassigned

where:

- *moduleName* is the name of the element type, such as "topic" or "map".
- *domainName* is the short name of the domain, for example, "highlight" or "utilities".
- *attrName* is the name of the specialized attribute, for example, "deliveryTarget".
- *ext* is the file extension, for example, "ent" or "mod".
- *qualifier* is a string that is specific to the constraints module and characterizes it, for example, "strict" or "requiredTitle" or "myCompany-".
- *Target* is the target of the constraint with an initial capital, for example, "Topic" or "HighlightDomain".

RELAX NG-based specialization modules

The DITA Technical Committee uses certain file-naming conventions for RNG-based specialization modules. While the grammar files shipped with DITA 2.0 do not include domain constraint or expansion modules, we suggest conventions for those modules also.

Module type	File name	Example
Structural	*moduleName*Mod.*rng*	conceptMod.rng
Element domain	*domainName*DomainMod.rng	highlightDomainMod.rng
Attribute domain	*attrName*AttDomain.rng	deliveryTargetAttDomain.rng
Constraint	*qualifierTarget*ConstraintMod.rng	strictTaskbodyConstraintMod.rng acmeHighlightDomainConstraintMod.mod **Comment by dstevens73** change mod to rng Kris Eberlein, 04 October 2022 Done **Disposition: Completed**
Expansion		sectionExpansionMod.rng cellPurposeAtt.rng acme-otherpropsAttExpansion.rng tableCellAttExpansion.rng

Comment by Kristen J Eberlein on 19 September 2022

The names of the expansion modules listed in the "Example" column are taken from the example topics. They do not follow a consistent pattern. I suspect that the same is true for file names used in the constraint example topics.

Also, is including "Mod" in element-domain or constraint files something we really want to do, or was it necessary for the RNG-to-DITA/XSD converter?

WEK: Adding "Mod" (and similar qualifiers) to the RNG filenames because all the RNG files must have the extension ".rng" whereas the DTD files can have different extensions.

Disposition: Unassigned

where:

- *moduleName* is the name of the element type, such as "topic" or "map".
- *domainName* is the short name of the domain, for example, "highlight" or "utilities".
- *attrName* is the name of the specialized attribute, for example, "deliveryTarget".
- *qualifier* is a string that is specific to the constraints module and characterizes it, for example, "strict" or "requiredTitle" or "myCompany-".
- *Target* is the target of the constraint with an initial capital, for example, "Topic" or "HighlightDomain".

G.2 Globally-unique identifiers in the base DITA edition

Each DITA grammar file has a globally-unique identifier. This identifier can reference either the latest version or a specific version of the grammar file.

Each of the following grammar files has globally-unique identifier:

- Document-type shell
- Structural vocabulary module
- Element- or attribute domain module
- Element-configuration module (constraint or expansion)

DTD-based grammar files

The public identifiers for the DTD files that are maintained by OASIS use the following format:

```
"-//OASIS//DTD DITA version information-type//EN"
```

where:

- *version* either is the specific version number (for example, 2.0) or 2.x, which represents the most recent version of DITA 2.x. Omitting the version number entirely is also equivalent to the most recent version of DITA 2.x.
- *information-type* is the name of the topic or map type, for example, Base Topic.

Note that "OASIS" is the owner identifier; this indicates that the artifacts are owned by OASIS. The keyword "DITA" is a convention that indicates that the artifact is DITA-related.

RNG-based grammar files

The URNs for the RNG files that are maintained by OASIS use the following format:

```
"urn:pubid:oasis:names:tc:dita:rng:information-type.rng:version"
```

where:

- *version* either is the specific version number (for example, 2.0) or 2.x, which represents the most recent version of DITA 2.x. Omitting the version number entirely is also equivalent to the most recent version of DITA 2.x.
- *information-type* is the name of the topic or map type, for example, basetopic

Note that "oasis" is the owner identifier; this indicates that the artifacts are owned by OASIS. The keyword "dita" is a convention that indicates that the artifact is DITA-related.

G.3 Domains provided in the base DITA edition

The base DITA edition includes a set of attribute- and element-domain specializations. The attribute domains are available for use in both maps and topics, while the element domains vary as to where they can be made available.

Attribute-specialization domains

The following table lists the attribute specializations that are included in the base DITA edition.

Domain	Description
`@audience`	Attribute for conditional processing based on target audience

Domain	Description
`@deliveryTarget`	Attribute for conditional processing based on target delivery mechanism
`@otherprops`	Attribute for conditional processing when an appropriate semantic is not developed
`@platform`	Attribute for conditional processing based on platform
`@product`	Attribute for conditional processing based on product

Element-domain specializations

The following table lists the element specializations that are included in the base DITA edition.

Comment by dstevens73
change element to element-domain

Kris Eberlein, 04 October 2022

Done

Disposition: Completed

Domain	Purpose	Where available	Short name
Alternative titles	Provides alternative titles for resources	Map & topic	alternativeTitles-d
DITAVALref	Enables filtering a branch of a DITA map	Map	ditavalref-d
Emphasis	Provides `<em>` and `<strong>` elements for indicating emphasis	Map & topic	emphasis-d
Hazard statement	Provides a hazard statement element that meets meets ANSI Z535 and ISO 3864 requirements	Map & topic	hazard-d
Highlighting	Provides typographic elements	Map & topic	hi-d
Map group	Provides convenience elements for use in DITA maps	Map	mapgroup-d
Utilities	Provides image maps and a sort key	Map & topic	ut-d

G.4 Document-type shells provided in the base DITA edition

The DITA specification contains a starter set of document-type shells. These document-type shells are commented and can be used as templates for creating custom document-type shells.

The following table lists the document-type shells that are included in the base DITA edition and the domains that are integrated into them.

Document-type shell	Domains included	Domains NOT included
Base map	All attribute domains The following element domains: • Alternative titles	Not applicable

Document-type shell	Domains included	Domains NOT included
	• DITAVAL reference • Emphasis • Hazard statement • Highlighting • Map group • Utilities	
Base topic	All attribute domains and the following element domains: • Alternative titles • Emphasis • Hazard statement • Highlighting • Utilities	The following element domains: • DITAVAL reference • Map group
Subject scheme	All attribute domains The following element domains: • Alternative titles • Emphasis • Highlighting	The following element domains: • DITAVAL reference • Hazard statement • Map group • Utilities