Darwin Information Typing Architecture (DITA) for Technical Communication Version 2.0

Working Draft 03

27 August 2019

This version:
OASIS has not yet provided the URLs for this document.

Previous version:
N/A

Latest version:
OASIS has not yet provided the URLs for this document.

Technical Committee:
OASIS Darwin Information Typing Architecture (DITA) TC

Chair:
Kristen James Eberlein (kris@eberleinconsulting.com), Eberlein Consulting LLC

Editors:
Name goes here
Name goes here

Additional artifacts:
This prose specification is one component of a work product that also includes:

- X (DITA source)
- X (Grammar files)

Related work:
...

Abstract:
The Darwin Information Typing Architecture (DITA) for Technical Communication Version 2.0 specification ...

Status:
This document was last revised or approved by the OASIS Darwin Information Typing Architecture (DITA) TC on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dita#technical.
TC members should send comments on this specification to the TC’s email list. Others should send comments to the TC’s public comment list, after subscribing to it by following the instructions at the “Send A Comment” button on the TC’s web page at https://www.oasis-open.org/committees/comments/index.php?wg_abbrev=dita.

This specification is provided under the RF on Limited Terms Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC’s web page (https://www.oasis-open.org/committees/dita/ipr.php).

Note that any machine-readable content (Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product’s prose narrative document(s), the content in the separate plain text file prevails.

Citation format:
When referencing this specification, the following citation format should be used:

[wp-abbrev]

_Darwin Information Typing Architecture (DITA) for Technical Communication Version 2.0._ Edited by <Add names here>. 27 August 2019. Working Draft 03. X. Latest version: X.
Notices

Copyright © OASIS Open 2019. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS’ procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.
# Table of contents

1 Introduction........................................................................................................................................6
   1.1 About DITA for Technical Communication..............................................................................6
      1.1.1 Written specification............................................................................................................6
      1.1.2 XML grammar files............................................................................................................7
   1.2 Terminology................................................................................................................................7
   1.3 IPR policy..................................................................................................................................7
   1.4 Normative references................................................................................................................8
   1.5 Non-normative references..........................................................................................................8
   1.6 Formatting conventions in the XHTML version of the specification........................................10
      1.6.1 Link previews......................................................................................................................10
      1.6.2 Navigation links..................................................................................................................11
2 Overview of DITA for Technical Communication.........................................................................13
   2.1 Topic and map specializations..................................................................................................13
      2.1.1 Bookmap............................................................................................................................13
      2.1.2 Concept................................................................................................................................14
      2.1.3 Glossary entry......................................................................................................................15
      2.1.4 Glossary group....................................................................................................................17
      2.1.5 Machinery task....................................................................................................................17
      2.1.6 Reference............................................................................................................................18
      2.1.7 General task.........................................................................................................................19
      2.1.8 Task....................................................................................................................................21
      2.1.9 Troubleshooting..................................................................................................................22
   2.2 Domain specializations..............................................................................................................26
      2.2.1 Equation and MathML domains...........................................................................................26
      2.2.2 Release management domain.............................................................................................27
      2.2.3 xNAL domain.......................................................................................................................28
3 Element reference.........................................................................................................................32
   3.1 Elements, A to Z.........................................................................................................................32
   3.2 Topic and map specializations..................................................................................................37
      3.2.1 Bookmap elements...............................................................................................................37
      3.2.2 Concept elements...............................................................................................................37
      3.2.3 Glossary elements.................................................................................................................64
      3.2.4 Reference elements..............................................................................................................66
      3.2.5 Task elements.......................................................................................................................78
      3.2.6 Troubleshooting elements....................................................................................................85
3.3 Domain specializations................................................................................................................101
   3.3.1 Equation domain..................................................................................................................101
   3.3.2 Markup domain.....................................................................................................................107
   3.3.3 MathML domain....................................................................................................................107
   3.3.4 Programming domain............................................................................................................111
   3.3.5 Release management domain................................................................................................124
   3.3.6 Software domain...................................................................................................................128
   3.3.7 SVG domain........................................................................................................................132
   3.3.8 User interface domain..........................................................................................................136
   3.3.9 XML mention domain...........................................................................................................139
   3.3.10 xNAL domain.....................................................................................................................143
4 Conformance..................................................................................................................................157
Appendix A Acknowledgments..........................................................................................................158
1 Introduction

The technical content part of DITA models the semantics of information types for technical communications.

The technical content part of DITA adds the semantics of technical-communication information-types to base DITA. This is done through several specializations:

- **Topic specializations:**
  - Concept
  - Glossary entry
  - Reference
  - Task
  - Troubleshooting

- **Domain specializations:**
  - Abbreviated form
  - Equation
  - Hazard statement
  - Highlight
  - Indexing
  - Markup
  - MathML
  - Programming
  - Release management
  - Software
  - SVG
  - User interface
  - Utilities
  - XML mention

- **Bookmap map specialization**

1.1 About DITA for Technical Communication

The DITA technical content specification is designed for users who use information typing and document complex applications and devices, such as software, hardware, medical devices, machinery, and more.

1.1.1 Written specification

The specification is written for implementers of the DITA standard, including tool developers and XML architects who develop specializations.

The specification contains several parts:

- Introduction
- Architectural specification
- Language reference
- Conformance statement
- Appendices

The specification is available in the following formats:
1.1.2 XML grammar files
The XML grammar files are available in RELAX NG (RNG), and XML Document-Type Definitions (DTD).
While the files should define the same DITA elements, the RELAX NG grammars are normative if there is a discrepancy.

1.2 Terminology
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMEND", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119].

MUST
This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT
This phrase, or the phrase "SHALL NOT", means that the definition is an absolute prohibition of the specification.

SHOULD
This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT
This phrase, or the phrase "NOT RECOMMENDED", means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

MAY
This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option must be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option must be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides).

1.3 IPR policy
This specification is provided under the RF on Limited Terms Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC’s web page (https://www.oasis-open.org/committees/dita/ipr.php).
1.4 Normative references

[RFC 2119]

[RFC 3986]

[RFC 5646]

[XML 1.0]

[XML 1.1]

1.5 Non-normative references

Non-normative references are references to external documents or resources that implementers of DITA might find useful.

[ciq-v3.0]
OASIS Committee Specification 02, Customer Information Quality Specifications Version 3.0. Name (xNL), Address (xAL), Name and Address (xNAL) and Party (xPIL), http://www.oasis-open.org/committees/download.php/29877/OASIS%20CIQ%20V3.0%20CS02.zip, 20 September 2008.

[ISO 8601]

[ISO/IEC 19757-3]

[MathML 3.0]

[NAMESPACES IN XML 1.0]

[NAMESPACES IN XML 1.1]
1.6 Formatting conventions in the XHTML version of the specification

Given the size and complexity of the specification, it is not generated as a single XHTML file. Instead, each DITA topic is rendered as a separate XHTML file.

The XHTML version of the specification uses certain formatting conventions to aid readers in navigating through the specification and locating material easily: Link previews and navigation links.

1.6.1 Link previews

The DITA specification uses the content of the DITA `<shortdesc>` element to provide link previews for its readers. These link previews are visually highlighted by a border and a colored background.

The link previews serve as enhanced navigation aids, enabling readers to more easily locate content. This usability enhancement is one of the ways in which the specification illustrates the capabilities of DITA and exemplifies DITA best practices.
The following screen capture illustrates how link previews are displayed in the XHTML version of the specification:

Figure 1: Link previews

2.2.1 DITA topics
DITA topics are the basic units of DITA content and the basic units of reuse. Each topic contains a single subject. Topics may be of specific specialized information types, such as task, concept, or reference, or may be generic, that is, without a specified information type.

2.2.2 DITA maps
This topic collection contains information about DITA maps and the purposes that they serve. It also includes high-level information about DITA map elements, attributes, and metadata.

2.2.3 Subject scheme maps and their usage
Subject scheme maps can be used to define controlled values and subject definitions. The controlled values can be bound to attributes, as well as element and attribute pairs. The subject definitions can contain metadata and provide links to more detailed information; they can be used to classify content and provide semantics that can be used in taxonomies and ontologies.

1.6.2 Navigation links
To ease readers in navigating from one topic to another, each XHTML file generated by a DITA topic contains navigation links at the bottom.

Parent topic
Takes readers to the parent topic, which is the topic referenced by the closest topic in the containment hierarchy

Previous topic
Takes readers to the previous topic in the reading sequence

Next topic
Takes readers to the next topic in the reading sequence

Return to main page
Takes readers to the place in the table of contents for the current topic in the reading sequence
The following screen capture illustrates how navigation links are displayed in the XHTML version of the specification:

**Figure 2: Navigation links**

When readers hover over the navigation links, the short description of the DITA topic also is displayed.
2 Overview of DITA for Technical Communication

The technical content architectural specification contains information about the technical-content document types and domains, as well as a topic that provides an overview of the DITA elements that support documenting troubleshooting information.

2.1 Topic and map specializations

The Technical Content package contains various document types: concept, reference, general task, strict task, machinery task, troubleshooting, glossary entry, and glossary group. These document types are designed specifically for information that describes how to use products and processes. The Technical Content package also includes the map document type.

2.1.1 Bookmap

The DITA bookmap specialization represents the key markup requirements for managing DITA content through book-oriented publication processes, including book metadata and book structures for organizing content.

The purpose of the bookmap specialization

Books and other printed media are popular ways to present DITA content. By specializing the general DITA map structure into the general structure and subject areas used by most book-oriented DTDs, bookmaps enable users to organize their DITA information into front matter, parts, chapters, and so forth. A rich set of metadata allows for recording information about the book, such as its authors and owners, versions, and production history.

The structure of the bookmap specialization

The `<bookmap>` element is the top-level element for a DITA bookmap. Most of the content for a bookmap is optional, allowing for specializations that further restrict the bookmap model.

A bookmap allows the following parts:

- An initial title or booktitle (booktitle has more semantics)
- Book metadata (publisher, author, copyright holders and dates, etc.)
- Front matter (placement for Table of Contents and other preliminary information)
- Any number of chapters or parts (parts can group chapters, chapters can group topics)
- An appendices section (similar to a part or a chapter, can group multiple appendices)
- Back matter (similar to front matter, notices, glossary, index, etc.)
- Relationship table

In book-oriented XML grammars, authors typically manage content structures as external entities, separate from the body of the book and referenced as embedded elements into the overall structure. Bookmap follows the same organizational approach, using the topicref-based structure of DITA maps as the archetype for the major divisions of a book.

Here is an example of a simple bookmap. It uses several mechanisms to include chapter content:

- Referencing a DITA map
- Referencing a DITA topic
• Nesting `<topicref>` elements

```xml
<bookmap id="taskbook">
  <booktitle>
    <mainbooktitle>Product tasks</mainbooktitle>
    <booktitlealt>Tasks and what they do</booktitlealt>
  </booktitle>
  <bookmeta>
    <author>John Doe</author>
    <bookrights>
      <copyrfirst>
        <year>2006</year>
      </copyrfirst>
      <bookowner>
        <person href="task_preface.dita">Jane Doe</person>
      </bookowner>
    </bookrights>
  </bookmeta>
  <chapter format="ditamap" href="installing.ditamap"/>
  <chapter href="configuring.dita"/>
  <chapter href="maintaining.dita">
    <topicref href="maintainstorage.dita"/>
    <topicref href="maintainserver.dita"/>
    <topicref href="maintaindatabase.dita"/>
  </chapter>
  <appendix href="task_appendix.dita"/>
</bookmap>
```

### 2.1.2 Concept

Concept topics are specialized from topic. They include the standard topic elements, including the short description, prolog, a body, and related links.

**The purpose of the concept information type**

Concepts provide background that helps readers understand essential information about a product, a task, a process, or any other conceptual or descriptive information. A concept might be an extended definition of a major abstraction such as a process or function. Conceptual information might explain the nature and components of a product and describe how it fits into a category of products. Conceptual information helps readers to map their knowledge and understanding to the tasks they need to perform and to provide other essential information about a product, process, or system.

**The structure of the concept topic**

The concept topic is specialized from the base topic information type. The top-level element for a DITA concept topic is the `<concept>` element. Every concept topic contains the standard topic elements, including title, short descriptions or abstract, prolog, a body, and related links.

The `<conbody>` element holds the main body-level elements of the concept topic. Like the `<body>` element of a base topic, the `<conbody>` allows paragraphs, lists, tables, figures and other general elements. It also provides two key elements that allow authors to subdivide the topic into parts, with or without titles. These subdivisions are called sections and examples. The `<conbody>` also allows `<div>`, `<bodydiv>`, and `<sectiondiv>` to facilitate grouping elements in the `<conbody>`.

**Limitations within `<conbody>`**

The `<conbody>` provides for an unlimited number of subdivisions in the form of sections and examples. However, once an author decides to incorporate a section or example in the `<conbody>`, only additional
sections or examples are allowed. Sections and examples can not nest, meaning that only one level of subdivision is permitted in the concept topic.

**Concept body primary subdivisions**

**<section>**
Represents an organizational division in a concept topic. Sections organize subsets of information within a larger topic. You can only include a simple list of peer sections in a topic; sections cannot be nested. A section can have an optional title.

**<example>**
Provides examples that illustrate or support the current topic. The `<example>` element has the same content model as `<section>`.

Following is an example of a simple concept topic. Note that once an example is used, it can be followed only by another example or by a section.

```
<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>
```

### 2.1.3 Glossary entry

Each glossary entry `<glossentry>` topic defines a single sense of one term. Besides identifying the term and providing a definition, the topic accommodates basic terminology information, such as part of speech. A glossentry topic might also include acronyms and acronym expansions. Glossentry topics can be assembled by authors or processes to create glossaries for various purposes, including books, websites, or other projects.

**The purpose of the glossary entry topic**

Defining terminology in a glossary ensures that a team of writers uses the same term for the same concept. A glossary added to a book or available online in conjunction with other subject matter provides the reader with definitions of unfamiliar terms and expands acronyms.

**The structure of the glossentry topic**

The top-level element for a DITA glossentry topic is the `<glossentry>` element. Every glossentry topic contains a `<glossterm>` and a `<glossdef>` element and optional `<related-links>`.

Where a term has multiple definitions, best practices call for the writer to create multiple glossentry topics with the same term in the `<glossterm>` element but different definitions in the `<glossdef>` element. A process can collate and group glossentry topics by term when generating formatted output. Note that definitions with the same term in one language can have different terms in other languages, so translations can result in different collation and grouping of the same set of glossentry topics.
Here is an example of a simple glossentry topic:

```xml
<glossentry id="ddl">
  <glossterm>Data Definition Language</glossterm>
  <glossdef>A language used for defining database schemas.</glossdef>
</glossentry>
```

To create a glossary, authors can group multiple entries together by
- authoring in a single document using the Glossary group document type
- authoring in a single document under a container topic using the database document type
- referencing the glossentry topics from a map
- using an automated process

For example, an automated process might assemble glossentry topics from a repository based on the `<term>` markup in a particular collection of topics.

### Acronyms defined within glossentry topics

The glossentry topic can be used to provide expansions of acronyms in online text and assist in the proper translation of acronyms into multiple languages. The acronym elements of the glossentry topic include the following:
- `<glossterm>` to enter the full text to which the acronym refers
- `<glossSurfaceForm>` to provide the appropriate rendering of the full text plus the acronym in each language
- `<glossAcronym>` to provide the acronym text itself

Here is an example of an acronym used in the glossentry topic:

```xml
<glossentry id="wmd" xml:lang="en">
  <glossterm>Weapons of Mass Destruction</glossterm>
  <glossBody>
    <glossSurfaceForm>Weapons of Mass Destruction (WMD)</glossSurfaceForm>
    <glossAlt>
      <glossAcronym>WMD</glossAcronym>
    </glossAlt>
  </glossBody>
</glossentry>
```

Here is an example of how the glossentry topic would be translated into Spanish:

```xml
<glossentry id="wmd" xml:lang="es">
  <glossterm>armas de destrucción masiva</glossterm>
  <glossBody>
    <glossSurfaceForm></glossSurfaceForm>
    <glossAlt>
      <glossAcronym></glossAcronym>
    </glossAlt>
  </glossBody>
</glossentry>
```

Note that because no acronym exists for the term in Spanish, the `<glossSurfaceForm>` and `<glossAcronym>` elements are left blank.

In some languages, the surface form that expands the acronym in its first use handles the formatting differently than in English. For example, in Polish, the acronym precedes the expansion.

```xml
<glossentry id="eu" xml:lang="pl">
  <glossterm>Unia Europejska</glossterm>
  <glossBody>
    <glossSurfaceForm>UE (Unia Europejska)</glossSurfaceForm>
  </glossBody>
</glossentry>
```
For more information about the correct use of acronym expansions in multiple languages, see Best Practice for Managing Acronyms and Abbreviations in DITA, produced by the DITA Translation Subcommittee. http://www.oasis-open.org/committees/download.php/29734/ AcronymBestPractice_08112008.doc

2.1.4 Glossary group
The glossary group (<glossgroup>) topic enables authors to include one or more glossary entry (<glossentry>) topics in a single collection file, rather than authoring each glossary entry topic in a separate file. The glossary group topic is a specialization of concept.

2.1.5 Machinery task
The machinery task document type supports the development of instructions for the completion of a procedure. The machinery task document type is built using the general task information type combined with the Machinery Taskbody Constraint.

The purpose of the machinery task information type
The machinery-task is designed to provide procedural information, similar to the strict task topic, and has a well-defined semantic structure that describes how to perform the steps required to accomplish a specific goal. Compared to the strict task information type, the machinery-task information type contains additional descriptive elements in the prelreqs section that add detail to the pre-requisites required to perform a task. The machinery-task topic is developed using the DITA constraint mechanism, in addition to specializations for new elements.

Machinery tasks are the essential building blocks to provide procedural information for machines, machinery equipment, assemblies, and apparatuses. A machinery-task information type answers the "How do I?" question by providing precise step-by-step instructions detailing the requirements that must be fulfilled, the actions that must be performed, and the order in which the actions must be performed. The machinery-task topic includes sections for describing the context, preliminary requirements, expected results, examples, closing requirements, and other aspects of a task.

The structure of the machinery-task topic
Similar to a strict DITA task, the <task> element is the top-level element for a machinery task topic. The machinery task document type contains a title and a taskbody with optional alternative titles (titlealts), a short description or abstract, a prolog, and related-links.

The <taskbody> element is the main body element inside a machinery-task topic. A machinery-task body has a very specific structure, with the following elements in this order: (<prelreqs> or <context> or <section>), <steps>, <result>, <example>, and <closereqs>. Each of the body sections is optional.

The machinery task includes two specialized element groups: <prelreqs> and <closereqs>. All other element groups are the same as the general task model.

<prelreqs>
The preliminary-requirements section of a task is used to describe what the user needs to know or do before starting the immediate task. The <prelreq> element is similar to the prerequisites section of the general task model but contains a more descriptive content model. The <prelreq> element
contains required conditions, required personnel, required equipment, supplies, spares, and safety
information.

<closereqs>
The close-requirements section is used to describe conditions that must be fulfilled after the
successful completion of the current task. It is often supported by links to the next task or tasks in the
<related-links> section. The <closereqs> element contains required conditions <reqconds>.

2.1.6 Reference
Reference topics are specialized from topic. They contain the standard topic elements, including title,
short descriptions or abstract, prolog, a body, and related links.

Purpose of the reference topic
Reference topics provide data that supports users as they perform a task. Reference topics might provide
lists and tables that include product specifications, parts lists, and other data that is often “looked up”
rather than memorized. A reference topic also can describe commands in a programming language or
required tools for a series of maintenance tasks.

Reference topics provide quick access to fact-based information. In technical information, reference
topics are used to list product specifications and parameters, provide essential data, and provide detailed
information on subjects such as the commands in a programming language. Reference topics can contain
any subject matter that has regular content, such as ingredients for food in recipes, bibliographic lists,
catalog items, and so on.

The structure of the reference topic
The top-level element for a reference topic is the <reference> element.

The <refbody> element contains the main body-level elements of the reference topic. Reference topics
limit the body to tables (both simple and complex), property lists, syntax sections, generic sections, and
examples.

All of the elements of <refbody> are optional; they can appear in any sequence and number.

Limitations on the reference body
The <refbody> provides for an unlimited number of subdivisions in the form of sections, examples,
syntax sections, property lists, and tables. Simple and complex tables can appear within sections,
examples, and syntax sections. They cannot appear within the property list or simple or complex table
sections. Sections, examples, syntax sections, table subdivisions, and property lists cannot nest; only one
level of subdivision is permitted in the reference topic.

The elements in the reference body
<section>
 Represents an organizational division in a reference topic. Sections organize subsets of information
within a larger topic. You can only include a simple list of peer sections in a topic; sections cannot be
nested. A section might have an optional title.

<refsyn>
 Contains syntax or signature content (for example, a command-line utility's calling syntax or an API's
signature). The <refsyn> contains a brief, possibly diagrammatic description of the subject's
interface or high-level structure.
<example>

Provides examples that illustrate or support the current topic. The <example> element has the same content model as <section>.
</example>

<table>

Organizes information according into a rows and columns. Table markup also allows for more complex structures, including spanning rows and columns, as well as table captions.
</table>

<simpletable>

Holds information in regular rows and columns and does not allow a caption.
</simpletable>

<properties>

Lists properties of a subject and their types, values, and descriptions.

Following is an example of a simple reference topic, including the <refsyn> element.

```xml
<reference id="boldproperty">
  <title>Bold property</title>
  <shortdesc>(Read-write) Whether to use a bold font for the specified text string.</shortdesc>
  <refbody>
    <refsyn>
      <synph>
        <var>object</var>.<delim>Font</delim>.<delim>KwD</delim>.<delim>Bold</delim> = <var>trueorfalse</var>
      </synph>
    </refsyn>
    <properties>
      <property>
        <proptype>Data type</proptype>
        <propvalue>Boolean</propvalue>
      </property>
      <property>
        <proptype>Legal values</proptype>
        <propvalue>True (1) or False (0)</propvalue>
      </property>
    </properties>
  </refbody>
</reference>
```

Following is an example of a simple reference topic, including the <property> element.

```xml
<reference id="oiltypes">
  <title>Oil Types</title>
  <shortdesc>The tables provide the recommended oil types.</shortdesc>
  <refbody>
    <properties>
      <prophead>
        <proptypehd>Oil type</proptypehd>
        <propvaluehd>Oil brand</propvaluehd>
        <propdeschd>Appropriate use</propdeschd>
      </prophead>
      <property>
        <proptype>Primary oil</proptype>
        <propvalue>AI1X</propvalue>
        <propdesc>Appropriate for one-cylinder engines</propdesc>
      </property>
      <property>
        <proptype>Secondary oil</proptype>
        <propvalue>B2Z</propvalue>
        <propdesc>Appropriate for two-cylinder engines</propdesc>
      </property>
    </properties>
  </refbody>
</reference>
```

2.1.7 General task

The general-task document type was introduced in DITA 1.2. It provides a less-strict content model for task-oriented information than was available previously. Some organizations might prefer the general-task
content model over the strict task model, as it can facilitate the migration of legacy content that does not follow the strict-task topic model. The general-task document type serves as the base for the strict-task and machine-industry-task document types. It also can be used to create new document types or specializations.

**The purpose of the general task information type**

Like the DITA strict task document type, the general task document and information types contain the essential building blocks to provide procedural information. Both task information types answer the "How do I?" question by providing step-by-step instructions that detail the requirements that must be fulfilled, the actions that must be performed, and the order in which the actions must be performed. Both task topics include sections for describing the context, prerequisites, expected results, and other aspects of a task.

The general task information type is specifically designed to accommodate task specializations that differ from the DITA task information type. It might also be used for the conversion of loosely structured tasks from other sources into DITA before they are restructured to follow the more restrictive DITA task model.

**The structure of the general task topic**

The `<task>` element is the top-level element for the general task topic. The general task topic contains a `<title>` and a `<taskbody>` with optional alternative titles (titlealts), a short description or `<abstract>`, a `<prolog>`, and `<related-links>`.

The following elements are described here because they are introduced as part of the general task model. All other elements are described in the strict task topic.

**<section>**

Represents an organizational division in a task topic. Sections organize subsets of information within the larger topic. Sections cannot be nested. A section can have an optional `<title>`.

**<steps-informal>**

Describes procedural task information that would not normally be ordered as `<steps>`, such as a group of general procedures that might all be applied in a particular situation. Instead of `<step>`, the `<steps-informal>` element uses `<ol>` and `<ul>` elements, which are less strictly defined than the `<step>` element. When converting legacy content, it might be simpler to convert numbered lists to `<ol>` elements than to `<step>` elements.

**Comparison of general and strict task**

The following table compares the structures of general and strict task:

<table>
<thead>
<tr>
<th>General taskbody</th>
<th>Strict taskbody constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>prerequisite (optional, in any order, any number)</td>
<td>prerequisite (optional, one only, must precede context)</td>
</tr>
<tr>
<td>context (optional, in any order, any number)</td>
<td>context (optional, one only, must follow prerequisite)</td>
</tr>
<tr>
<td>section (optional, in any order, any number)</td>
<td>(not defined for strict taskbody)</td>
</tr>
<tr>
<td>steps</td>
<td>steps</td>
</tr>
<tr>
<td>steps-unordered</td>
<td>steps-unordered</td>
</tr>
<tr>
<td>steps-informal</td>
<td>(not defined for strict taskbody)</td>
</tr>
<tr>
<td>result (optional, one only, precedes example)</td>
<td>result (optional, one only, precedes example)</td>
</tr>
</tbody>
</table>
2.1.8 Task

The strict task-document type supports the development of instructions for the completion of a procedure. The strict-task document type is built using the general-task information type combined with the strict taskbody constraint. See the reference below to ensure that you have the correct task document type when you update to DITA 1.3 from DITA 1.0 or DITA 1.1.

The purpose of the standard task information type

Tasks are the essential building blocks to provide procedural information. A task information type answers the “How do I?” question by providing precise step-by-step instructions detailing the requirements that must be fulfilled, the actions that must be performed, and the order in which the actions must be performed. The <task> topic includes sections for describing the context, prerequisites, expected results, and other aspects of a task.

The structure of the <task> topic

The <task> element is the top-level element for the strict task topic. The strict task document type contains a <title> and a <taskbody> with optional alternative titles (<titlealts>), a short description or <abstract>, a <prolog>, and <related-links>.

The <taskbody> element is the main body element inside a strict task document type. The strict task body has a constrained structure, with these optional elements in the following order:

<prereq>
Describes information that the user needs to know or do before starting the immediate task. This section can occur only once.

<context>
Provides background information for the task. This information helps the users understand the purpose of the task and what they will gain by completing the task correctly. This section should be brief and does not replace or recreate a concept topic on the same subject, although the context section might include some conceptual information. This section can occur only once.

<steps>
Provides the main content of the task topic. A <task> consists of a series of steps that accomplish the task. The <steps> element must have one or more <step> elements, which provide the specifics about each step in the task. The <steps> element can occur only once.

The <step> element represents an action that a user must follow to accomplish a task. Each <step> in a <task> must contain a command <cmd> element which describes the particular action the user must perform to accomplish the overall task. The <step> element can also contain information <info>, tutorial information <tutorialinfo>, a step example <stepxmp>, choices <choices>, a step result <stepresult>, or troubleshooting <steptroubleshooting>, although these are optional.

<steps-unordered>
Provides alternative content for the <task> topic, allowing for a single step in a procedure or a set of commands that need not be performed in a specific order.
Describes the expected outcome for the task as a whole.

Describes actions that a user might take if the task does not produce the expected results.

Provides an example that illustrates or supports the task.

Describes steps or tasks that the user should do after the successful completion of the current task. It is often supported by links to the next task or tasks in the <related-links> section.

Here is an example of a task topic:

```
<task id="birdhousebuilding">
  <title>Building a bird house</title>
  <shortdesc>Building a birdhouse is a perfect activity for adults to share with their children or grandchildren. It can be used to teach about birds, as well as the proper use of tools.</shortdesc>
  <taskbody>
    <prereq>To build a sound birdhouse, you will need a complete set of tools: 
    <ul>
      <li>hand saw</li>
      <li>hammer ... </li>
    </ul>
    <context>Birdhouses provide safe locations for birds to build nests and raise their young. They also provide shelter during cold and rainy spells.</context>
    <steps>
      <step><cmd>Lay out the dimensions for the birdhouse elements.</cmd></step>
      <step><cmd>Cut the elements to size.</cmd></step>
      <step><cmd>Drill a 1 1/2" diameter hole for the bird entrance on the front.</cmd>
      <info>You need to look at the drawing for the correct placement of the hole.</info></step>
    </steps>
    <result>You now have a beautiful new birdhouse!</result>
    <postreq>Now find a good place to mount it.</postreq>
  </taskbody>
</task>
```

Maintaining specializations using the strict task model

Organizations that have created specializations based on the DITA 1.0 and 1.1 strict task model can review the recommendations in STUB CONTENT for information about how to maintain their specializations.

2.1.9 Troubleshooting

The troubleshooting topic type provides markup for corrective action information such as troubleshooting and alarm clearing.

The troubleshooting information type

In its simplest form, troubleshooting information follows this pattern:

1. A condition or symptom. Usually the condition or symptom is an undesirable state in a system, a product, or a service that a reader wants to correct.
2. A cause for the condition or symptom.
3. A remedy for the condition or symptom.

The troubleshooting topic provides sections for describing the condition, causes, and remedies needed to restore a system, a product, or a service to normal.
For some conditions there could be more than one cause-remedy pair. The troubleshooting topic accommodates this. Typically, a cause is immediately followed by its remedy. Multiple cause-remedy pairs can provide a series of successive fall-backs for resolving a condition.

Cause and remedy might occur in combinations other than pairs. It is possible to have:

- Multiple causes with the same remedy
- A single cause with more than one remedy
- A remedy with no known cause
- A cause with no known remedy

The troubleshooting information type also can be used to document alarm clearing strategies.

**The structure of the troubleshooting topic**

The top-level element for troubleshooting topics is `<troubleshooting>`. The `<troubleshooting>` element contains a `<title>` with optional alternative titles (`<titlealts>`), a short description or `<abstract>`, a `<prolog>`, a `<troublebody>`, and `<related-links>`.

`<troublebody>` is the main body element in a troubleshooting topic. The `<troublebody>` element contains the following elements:

`<condition>`

This optional element is the first child of `<troublebody>`, and it describes a condition or symptom that is associated with an undesirable state in a system, a product, or a service. In cases where the topic title fully explains the condition, do not use this element.

`<troubleSolution>`

One or more `<troubleSolution>` elements must appear in the `<troublebody>` element. `<troubleSolution>` is a wrapper element for `<cause>` and `<remedy>`, each of which are a cause-remedy pair.

The `<troubleSolution>` element contains the following elements:

`<cause>`

This optional, repeatable, first-child of `<condition>` `<troubleSolution>` describes a possible cause for the condition.

`<remedy>`

This optional, repeatable, last-child of `<troubleSolution>` describes a possible remedy for the condition.

The `<remedy>` element begins with an optional `<title>` element followed by an optional `<responsibleParty>` element followed by either a `<steps>` element, a `<steps-unordered>` element, or a `<steps-informal>` element. The content models for `<steps>`, `<steps-unordered>`, and `<steps-informal>` are borrowed from `<task>`. This allows remedy to reuse steps from tasks.

`<responsibleParty>`

This optional first child of `<remedy>` indicates who is expected to perform the steps that are outlined in the `<remedy>` element.

Here is an example of a troubleshooting topic:

```xml
<troubleshooting id="nologon">
<title>Cannot log on</title>
<shortdesc>Login attempts have failed</shortdesc>
<troublebody>
```
The system does not accept your login credentials.

The CapsLock key might be on.

Verify that the CapsLock key is off.

The password that you are using does not match the one that is stored in the system.

Go to Account management, and follow the instructions.

The account name you are using does not match the one stored in the system.

Still cannot log on

If none of the previous solutions work, consider asking for help. Contact your system administrator if your organization has one; otherwise, contact our support team.

Related concepts

2.2.1.9.1 Troubleshooting information (24)

Troubleshooting information is invaluable to technical content users because it provides advice to correct problems with task performance.

2.1.9.1 Troubleshooting information

Troubleshooting information is invaluable to technical content users because it provides advice to correct problems with task performance.

User studies show that users find troubleshooting information invaluable because they frequently access technical information only when they have encountered a problem in performing a task.
The DITA troubleshooting elements added in DITA 1.3 encourage authors to provide troubleshooting information directly within a task, at the point of need, and to label the troubleshooting information appropriately. Minimalism research indicates that users are most likely to read troubleshooting information if it is visibly labeled. We recommend that implementations use style sheets that add symbols or text to draw attention to the troubleshooting elements in the topics.

Troubleshooting information can be added to an individual step in a DITA task; it also can be added after the steps are completed. Troubleshooting information can be added in any topic type in a `<note>` element with the `@type` attribute set to “trouble”. However, we recommend that users use either `<steptroubleshooting>` or `<tasktroubleshooting>` when working on a task topic, rather than inserting an `<note>` element with the `@type` attribute set to “trouble”.

The examples below illustrate the three options.

**Step troubleshooting**

Troubleshooting information can be added following a step in a procedure if it is likely that the user will encounter a problem in performing the step.

The `<steptroubleshooting>` element can occur following the `<cmd>` element in the `<step>` or `<substep>` element. The `<steptroubleshooting>` element ends the `<step>` or `<substep>` element. Another element, such as `<info>` or `<stepxmp>`, cannot be added after the `<steptroubleshooting>` element.

The first example shows troubleshooting information added to a step. The second example shows the troubleshooting information following a step result.

```xml
<step>
  <cmd>Select <uicontrol>Shut Down</uicontrol> from the <uicontrol>File</uicontrol> Menu.</cmd>
  <steptroubleshooting>If a problem with the computer prevents you from choosing Shut Down—for example, if the computer “freezes” so that the pointer does not respond to the trackpad—you can turn off the computer by holding down the Control, Option, Command, and Power On keys at the same time.</steptroubleshooting>
</step>

<step>
  <cmd>Select the element for which you want to assign a conditional-processing attribute, and, in the <wintitle>Attributes</wintitle> window, select the conditional-processing attribute.</cmd>
  <stepresult>The permissible values for the attribute are displayed in a drop-down list.</stepresult>
  <steptroubleshooting>If the list of controlled values is not displayed, ensure that the root map is opened in the <wintitle>DITA Maps Manager</wintitle> window.</steptroubleshooting>
</step>
```

**Task troubleshooting**

Troubleshooting information can be added after the procedural component to assist the user in correcting a problem that might have occurred. The `<tasktroubleshooting>` element is one of four optional elements that can follow the `<steps>` element. When these optional element are used, they must appear in the following order: `<results>`, `<tasktroubleshooting>`, `<example>`, and `<postreq>`.

```xml
<steps>
  <step>
    <cmd>When the fast blinking stops, press ....</cmd>
  </step>
</steps>
```
<uicontrol>small espresso</uicontrol>.</cmd>
</step>
<tasktroubleshooting>If the <uicontrol>small espresso</uicontrol> button is not lit, recycle the unit by turning the external <uicontrol>Power</uicontrol> off and on.
</tasktroubleshooting>

**Troubleshooting note**

A `<note>` element can include `type="trouble"` to indicate that the note deals with a potential problem that the user might encounter. We recommend that DITA implementations configure their CSS or stylesheets to include appropriate text or an icon in the generated output.

Best practices for authoring DITA content mandate that the troubleshooting note should not be used in place of `<steptroubleshooting>` or `<tasktroubleshooting>` in a task topic.

```xml
<concept>
  <p>If you expose your camera to sudden changes in temperature or humidity, you might experience some condensation in the camera. Avoid the possibility of condensation because it might result in soil on the lens or the monitor, cause mold, or damage the camera.</p>
  <note type="trouble">If you do get condensation, turn off the camera and wait about two hours before using it. Once the camera adjusts to the surrounding temperature, the fogging will clear naturally.</note>
</concept>
```

Related concepts

2.2.1.9 Troubleshooting (22)
The troubleshooting topic type provides markup for corrective action information such as troubleshooting and alarm clearing.

### 2.2 Domain specializations

This section contains information about the technical content domains, including a general overview and more specific information about the xNAL, release management, equation, and MathML domains.

#### 2.2.1 Equation and MathML domains

The MathML and equation domains provide general containers for equations (inline, block, and display) and an integration with the MathML standard. The equation containers identify equations separate from the data that provides the equation content, which might take many forms, including MathML markup, images, and non-XML formats like TeX and ASCII equations.

These two domains provide support for mathematics elements:

**MathML domain**

Provides the `<mathml>` element, which is a specialization of `<foreign>`. The `<mathml>` element can contain direct MathML markup, or it can use the `<mathmlref>` element to include MathML markup that is stored in a separate, non-DITA document. When this domain is integrated without additional constraints, the `<mathml>` element is allowed wherever `<foreign>` is allowed.

The equation domain elements can be used to distinguish equations semantically. While authors can include the `<mathml>` element directly in content, the recommended best practice is to always
include `<mathml>` within one of the equation domain elements or their semantic equivalent in a non-OASIS-defined DITA vocabulary.

**Equation domain**

Provides the `<equation-inline>`, `<equation-block>`, and `<equation-figure>` elements. This domain can be used independently of the MathML domain. The elements in the equation domain enable authors to clearly distinguish equations from other type of content; these markup distinctions can enable formatting distinctions, numbering of equations, and more.

### 2.2.2 Release management domain

The release management domain provides markup that enables content developers to log information about changes that have been made to a DITA topic or map. This information can be used to generate release notes, which can help users locate significant changes in revisions of large documents.

Processors can gather the information and assemble it into an appendix or separate document, thus eliminating the need for separate release-note topics or external tracking mechanisms such as spreadsheets or databases. Regulated industries can use the release management domain to provide the records that are required by regulatory agencies.

This figure shows the structure of the release management elements.

**Figure 3: Release management elements**

The following list provides a brief description of the element:

- `<change-historylist>`
  - Contains `<change-item>` elements, each of which represents a release note, which is information about a single change

- `<change-item>`
  - Contains information about a release note

- `<change-person>`
  - Contains the name of a person who made the change

- `<change-organization>`
  - Contains the name of the organization that requested the change
<change-revisionid>
Contains an identifier that is associated with the change
</change-revisionid>

<change-request-reference>
Contains information that provides traceability to an external change request or other ticketing system
</change-request-reference>

<change-request-system>
Contains the name of the external system that is referenced
</change-request-system>

<change-request-id>
Contains an identifier that is associated with the change request, such as an issue ID or ticket number
</change-request-id>

<change-started>
Contains a string that holds date and time information about when the change was started
</change-started>

<change-completed>
Contains a string that holds date and time information about when the change was finished
</change-completed>

<change-summary>
Contains a descriptive summary of the change
</change-summary>

This figure shows three simple change-items added to a single topic. This topic is used in documentation for two products, A and B.

Figure 4: Excerpt from topic prolog

<prolog>
<!--...
<change-historylist>
<!--
<change-item product="productA productB">
  <change-person>Joan Smith</change-person>
  <change-completed>2013-03-23</change-completed>
  <change-summary>Made change 1 to both products</change-summary>
  <data>Details of change 1</data>
</change-item>
<!--
<change-item product="productA">
  <change-person>Bill Carter</change-person>
  <change-completed>2013-06-07</change-completed>
  <change-summary>Made change 2 to product A</change-summary>
  <data>Details of change 2</data>
</change-item>
<change-item product="productA productB">
  <change-person>Richard Smith</change-person>
  <change-completed>2013-07-20</change-completed>
  <change-summary>Made change 3 to both products</change-summary>
  <data>Details of change 3</data>
</change-item>
<!--
</change-historylist>
<!--
</prolog>

2.2.3 xNAL domain
The DITA xNAL domain specialization defines a number of metadata elements and attributes that are useful in representing personal/organizational names and addresses. The metadata can be used to identify authors and content owners. The OASIS xNAL Standard (extensible Name and Address Language) was selected to represent close mappings from the DITA bookmap metadata content model to an existing standard. xNAL is included in the Bookmap and the LearningBookmap document types.

The OASIS Customer Information Quality (CIQ) standard for global-customer information management contains the definition of the OASIS extensible Name and Address Language (xNAL) metadata elements. Version 2 of the standard states:

  The objective of xNAL is to describe a common structure for Personal/Organization Names and Addresses that would enable any applications that want to represent customer names and
addresses in a common standard format. The applications could be CRM/e-CRM, Customer Information Systems, Data Quality (Parsing, Matching, Validation, Verification, etc.), Customer Data Warehouses, Postal services, etc.

However, any party for its own purposes and applications can use xNAL grammar or parts of it. The DITA xNAL specialization is based on the OASIS extensible Name and Address Language metadata elements. Due to differences between the two processing architectures, the DITA xNAL domain does not incorporate all of the definitions from the OASIS xNAL standard directly. Instead, there is a transformational equivalence between the DITA and OASIS xNAL definitions for names and addresses. This equivalence enables XML-aware tools in workflow systems to capture and manipulate names and addresses in a standard manner.

The xNAL domain is available for use in the <bookmap> document type. It can be included in any DITA document types that require metadata for names and addresses.

2.2.3.1 xNAL usage guidelines

Extended information and usage examples for DITA bookmap metadata elements associated with OASIS xNAL Standard (extensible Name and Address Language).

DITA bookmaps use a set of elements associated with a subset of the OASIS extensible Name and Address Language (xNAL) specification (Version 2) to denote name and address information related to persons and organizations.

While the elements share element names, and the expectation is that content written using this type of metadata is be straightforward to transform, the element name pairs do not share content models. The difference in content models reflects the different purposes of the two standards. The purpose of the name and address elements in DITA is to identify persons or organizations associated with the creation of a document; the purpose of the name and address elements in xNAL is to support customer resource management.

The examples shown after the table provide sample tagging methods for name and address information, using the DITA elements associated with xNAL.

DITA elements associated with xNAL elements

The set of bookmap elements associated with elements from the OASIS extensible Name and Address Language (xNAL) standard are listed in the table below.

Table 1: DITA elements associated with xNAL elements

<table>
<thead>
<tr>
<th>&lt;addressdetails&gt;</th>
<th>&lt;honorific&gt;</th>
<th>&lt;otherinfo&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;administrativearea&gt;</td>
<td>&lt;lastname&gt;</td>
<td>&lt;person&gt;</td>
</tr>
<tr>
<td>&lt;authorinformation&gt;</td>
<td>&lt;locality&gt;</td>
<td>&lt;personinfo&gt;</td>
</tr>
<tr>
<td>&lt;contactnumber&gt;</td>
<td>&lt;localityname&gt;</td>
<td>&lt;personname&gt;</td>
</tr>
<tr>
<td>&lt;contactnumbers&gt;</td>
<td>&lt;middlename&gt;</td>
<td>&lt;postalcode&gt;</td>
</tr>
<tr>
<td>&lt;country&gt;</td>
<td>&lt;namedetails&gt;</td>
<td>&lt;thoroughfare&gt;</td>
</tr>
<tr>
<td>&lt;emailaddress&gt;</td>
<td>&lt;organization&gt;</td>
<td>&lt;url&gt;</td>
</tr>
<tr>
<td>&lt;emailaddresses&gt;</td>
<td>&lt;organizationinfo&gt;</td>
<td>&lt;urls&gt;</td>
</tr>
<tr>
<td>&lt;firstname&gt;</td>
<td>&lt;organizationname&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;generationidentifier&gt;</td>
<td>&lt;organizationnamedetails&gt;</td>
<td></td>
</tr>
</tbody>
</table>
Example 1: Tagging personal information in DITA
This example shows a way to tag the following personal name and description.

Mr. Ram V. Kumar Jr.
Chief Technologist
MSI Business Solutions

<authorinformation>
  <personinfo>
    <namedetails>
      <personname>
        <honorific>Mr.</honorific>
        <firstname>Ram</firstname>
        <middlename>V.</middlename>
        <lastname>Kumar</lastname>
        <generationidentifier>Jr.</generationidentifier>
        <otherinfo>Chief Technologist</otherinfo>
      </personname>
    </namedetails>
  </personinfo>
  <organizationinfo>
    <namedetails>
      <organizationnamedetails>
        <organizationname>MSI Business Solutions</organizationname>
      </organizationnamedetails>
    </namedetails>
  </organizationinfo>
</authorinformation>

Example 2: Tagging address information in DITA
This example shows a way to tag the following address.

23 Archer St.
Chatsworth
NSW 2067
Australia

<addressdetails>
  <thoroughfare>123 Archer St.</thoroughfare>
  <locality>
    <localityname>Chatsworth</localityname>
    <postalcode>2067</postalcode>
  </locality>
  <administrativearea>NSW</administrativearea>
  <country>Australia</country>
</addressdetails>

Example 3: Tagging complex name and address information in DITA
This example shows two ways to tag a fairly complex collection of personal, organizational, and address information.

Mr. Samuel L. Johnson Jr.
Chief Technologist
c/o XYZ Corporation
52 New Main St.
Carrboro, NC 27510 USA
email: johnson@example.com
phone: 919-555-7987
This method tags all the organizational information as associated with the identified person.

```xml
<personinfo>
  <namedetails>
    <personname>
      <firstname>Samuel</firstname>
      <middlename>L.</middlename>
      <lastname>Johnson</lastname>
      <generationidentifier>Jr.</generationidentifier>
      <otherinfo>Chief Technologist</otherinfo>
      <otherinfo>c/o XYZ Corporation</otherinfo>
    </personname>
  </namedetails>
  <addressdetails>
    <thoroughfare>52 New Main St.</thoroughfare>
    <locality>
      <localityname>Carrboro</localityname>
      <postalcode>27510</postalcode>
    </locality>
    <administrativearea>NC</administrativearea>
    <country>USA</country>
  </addressdetails>
  <contactnumbers>
    <contactnumber>919-555-7987</contactnumber>
  </contactnumbers>
  <emailaddresses>
    <emailaddress>johnson@example.com</emailaddress>
  </emailaddresses>
</personinfo>
```

The following method separates the person and organization information. It might be used if it were necessary to associate address information with organizations rather than persons.

```xml
<authorinformation>
  <personinfo>
    <namedetails>
      <personname>
        <firstname>Samuel</firstname>
        <middlename>L.</middlename>
        <lastname>Johnson</lastname>
        <generationidentifier>Jr.</generationidentifier>
        <otherinfo>Chief Technologist</otherinfo>
      </personname>
    </namedetails>
    <contactnumbers>
      <contactnumber>919-555-7987</contactnumber>
    </contactnumbers>
    <emailaddresses>
      <emailaddress>johnson@example.com</emailaddress>
    </emailaddresses>
  </personinfo>
  <organizationinfo>
    <namedetails>
      <organizationname>XYZ Corporation</organizationname>
      <otherinfo>c/o</otherinfo>
    </namedetails>
    <addressdetails>
      <thoroughfare>52 New Main St.</thoroughfare>
      <locality>
        <localityname>Carrboro</localityname>
        <postalcode>27510</postalcode>
      </locality>
      <administrativearea>NC</administrativearea>
      <country>USA</country>
    </addressdetails>
  </organizationinfo>
</authorinformation>
```
3 Element reference

This section contains topics for each element defined in the technical content specializations. These elements include the original concept, task, and reference specializations, as well as specializations added in later releases. It also includes domains designed primarily for technical content.

3.1 Elements, A to Z

This section provides an alphabetized list of links to all elements in the specification.

abbreviated-form (75)
abbrevlist (38)
addressdetails (144)
administrativearea (144)
amendments (39)
apiname (111)
appendices (39)
appendix (40)
approved (51)
authorinformation (143)
backmatter (40)
bibliolist (41)
bookabstract (41)
bookchangehistory (52)
bookevent (52)
bookeventtype (53)
bookid (53)
booklibrary (42)
booklist (42)
booklists (43)
bookmark (37)
bookmeta (50)
booknumber (53)
bookowner (54)
bookpartno (54)
bookrestriction (55)
bookrights (55)
booktitle (43)
booktitlealt (43)
cause (100)
change-completed (124)
change-historylist (125)
change-item (125)
change-organization (126)
change-person (126)
change-request-id (126)
3.2 Topic and map specializations

Content TBD

3.2.1 Bookmap elements

Elements in the bookmap section are used to organize DITA content into book form. They include elements for dividing up content, such as chapter and appendix, as well as metadata specific to publishing.

3.2.1.1 Bookmap content elements

The Bookmap specialization of ditamap supports standard book production for collections of DITA topics.

The OASIS document type for the bookmap specialization also includes substantial book metadata for describing authors, based on the eXtensible Name and Address Language, or xNAL.

3.2.1.1.1 <bookmap>

The <bookmap> element is a map specialization used to describe the relationships among a set of DITA topics intended to be configured as a traditional book.

Usage information

Bookmaps consist of references to topics organized as book content. The topic references therefore are labeled according to the book components they point to, such as book title, front matter, chapter, and appendix.
Specialization hierarchy
- map/map bookmap/bookmap

Attributes

Example

```xml
<bookmap xml:lang="en-us">
  <booktitle>
    <booklibrary>Books about stuff</booklibrary>
    <mainbooktitle>A book about one thing</mainbooktitle>
  </booktitle>
  <bookmeta>
    <bookrights>
      <copyrfirst><year>1994</year></copyrfirst>
      <copyrlast><year>2006</year></copyrlast>
      <bookowner>OASIS</bookowner>
    </bookrights>
  </bookmeta>
  <frontmatter>
    <booklists>
      <toc/>
      <figurelist/>
      <tablelist/>
    </booklists>
    <bookabstract href="MyBookAbstract.dita"/>
    <preface href="preface.dita"/>
  </frontmatter>
  <chapter href="chapter1.dita">
    <topicref href="subchap1.dita"/>
  </chapter>
  <chapter href="chapter2.dita">
    <topicref href="subchap2.dita"/>
  </chapter>
  <appendix href="app1.dita">
    <topicref href="insideApp1.dita"/>
  </appendix>
  <appendix href="app2.dita">
    <topicref href="insideApp2.dita"/>
  </appendix>
  <backmatter>
    <amendments href="updatesToTheBook.dita"/>
    <booklists>
      <trademarklist href="listoftrademarks.dita"/>
      <indexlist/>
    </booklists>
  </backmatter>
</bookmap>
```

3.2.1.1.2 `<abbrevlist>`
The `<abbrevlist>` element references a list of abbreviations.

Usage information

This element indicates to the processing software that the author wants an abbreviation list generated at the particular location.
**Specialization hierarchy**
- map/topicref bookmap/abbrevlist

**Attributes**
STUB CONTENT

**Example**
```
<abbrevlist href="abbrev.dita"/>
```

**3.2.1.1.3 <amendments>**
The `<amendments>` element references a list of amendments or updates to the book.

**Formatting expectations**
The `<amendments>` element indicates to the processing software that the author wants an amendments list generated at the particular location.

**Specialization hierarchy**
- map/topicref bookmap/amendments

**Attributes**
STUB CONTENT

**Example**
See the example for `bookmap` (37).

**3.2.1.1.4 <appendices>**
The `<appendices>` element is an optional wrapper for `<appendix>` elements within a bookmap.

**Specialization hierarchy**
- map/topicref bookmap/appendices

**Attributes**
STUB CONTENT

**Example**
```
<appendices toc="yes" print="no">
  <topicmeta>
    <navtitle>Appendices</navtitle>
  </topicmeta>
  <appendix href="return-codes.dita"/>
  <appendix href="messages.dita"/>
  <appendix href="extra-info.dita"/>
</appendices>
```
3.2.1.1.5 <appendix>
The <appendix> element references a topic as a appendix within a book.

**Specialization hierarchy**
- map/topicref bookmap/appendix

**Attributes**
STUB CONTENT

**Example**
Appendix topics that include subtopics:

```xml
<appendix href="intro.dita">
  <topicref href="caring.dita"/>
  <topicref href="feeding.dita"/>
</appendix>
<appendix href="setup.dita">
  <topicref href="prereq.dita"/>
  <topicref href="download.dita"/>
</appendix>
```

Appendix that references a ditamap of content:

```xml
<appendix href="intro.ditamap" format="ditamap"/>
```

3.2.1.1.6 <backmatter>
The <backmatter> element contains the material that follows the main body of a document and any appendices.

**Usage information**
Backmatter might include items such as a colophon, legal notices, and various types of book lists such as a glossary or an index.

**Specialization hierarchy**
- map/topicref bookmap/backmatter

**Attributes**
STUB CONTENT

**Example**
See the example in bookmap (37).
3.2.1.7 <bibliolist>
The <bibliolist> element references a topic containing a list of bibliographic entries within the book.

Processing expectations
A <bibliolist> indicates to the processing software that the author wants a bibliography, containing links to related books, articles, published papers, or other types of material, generated at the particular location. If no @href attribute is specified on the <bibliolist> element, an external processor might generate a list of bibliographic entries at this location.

Specialization hierarchy
- map/topicref bookmap/bibliolist

Attributes
STUB CONTENT

Example

```xml
<bookmap>
  <!-- ... -->
  <backmatter>
    <amendments href="updatesToTheBook.dita"/>
    <booklists>
      <trademarklist href="listoftrademarks.dita"/>
      <bibliolist href="bibliography.dita"/>
      <indexlist/>
    </booklists>
  </backmatter>
</bookmap>
```

3.2.1.8 <bookabstract>
The <bookabstract> element references a topic used within a bookmap as a brief summary of book content

Usage information
A bookabstract is generally output as part of the book's front matter. It is used to help the reader quickly evaluate the book's purpose.

Specialization hierarchy
- map/topicref bookmap/bookabstract

Attributes
STUB CONTENT

Example
See the example in bookmap (37).
3.2.1.9 <booklibrary>
The <booklibrary> element contains the library information for a book.

Usage information
Library entries contain information about the series, library, or collection of documents to which the book belongs.

Specialization hierarchy
- topic/ph bookmap/booklibrary

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
See the example in bookmap (37).

3.2.1.10 <booklist>
The <booklist> element is a general purpose element, designed for use in specializations, that references a topic or map containing a list of items within the book.

Usage information
The <booklist> element is a general purpose element, designed for use in specializations, that references a topic or map containing a list of items within the book. For example, it could be used to reference a topic that contains a list of authors for the book. When a more specific element is already available, such as <tablelist> for a list of tables, use that element instead.

Specialization hierarchy
- map/topicref bookmap/booklist

Attributes
STUB CONTENT

Example
In this case the <booklist> element references a topic that contains a list of authors of topics in this document.

```xml
<booklists>
  <toc/>
  <tablelist/>
  <booklist href="authors.dita" navtitle="List of authors"/>
</booklists>
```
3.2.1.11 <booklists>

The `<booklists>` element references lists of various kinds within the book.

**Usage information**

Booklists can be used within `<frontmatter>` to reference a `<toc>`, `<tablelist>`, and `<figurelist>`, or within `<backmatter>` to reference a `<glossarylist>`, `<indexlist>`, and `<abbrevlist>`. It indicates to the processing software that the author wants the lists generated at the `<booklists>` location.

**Specialization hierarchy**

- map/topicref bookmap/booklists

**Attributes**

The following attributes are available on this element: `STUB CONTENT`, `STUB_CONTENT` (with a narrowed definition of `@chunk` given below), `STUB_CONTENT`, and `@keyref`. This element also uses `@scope`, `@format`, and `@type` from `STUB_CONTENT`.

**Example**

See the example in `bookmap` (37).

3.2.1.12 <booktitle>

The `<booktitle>` element contains the title information for a book, including the library title, main title, subtitle, and other titles (as required).

**Specialization hierarchy**

- topic/title bookmap/booktitle

**Attributes**

The following attributes are available on this element: `STUB CONTENT` (without the Metadata attribute group) and `#unique_24/unique_24_Connect_42_outputclass`.

**Example**

See the example in `bookmap` (37).

3.2.1.13 <booktitlealt>

The `<booktitlealt>` element contains the alternative title, subtitle, or short title for a book.

**Usage information**

Booktitlealt can be specialized into a specific element for those or other purposes.

**Specialization hierarchy**

- topic/ph bookmap/booktitlealt
Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example

```xml
<bookmap>
  <booktitle>
    <mainbooktitle>This is my big and fancy book</mainbooktitle>
    <booktitlealt>Shorter title</booktitlealt>
  </booktitle>
</bookmap>
```

3.2.1.14 <chapter>
The <chapter> element references a topic or map as a chapter within a book.

Specialization hierarchy
- map/topicref bookmap/chapter

Attributes
STUB CONTENT

Example

Chapter topics that include subtopics:

```xml
<chapter href="intro.dita">
  <topicref href="caring.dita"/>
  <topicref href="feeding.dita"/>
</chapter>
<chapter href="setup.dita">
  <topicref href="prereq.dita"/>
  <topicref href="download.dita"/>
</chapter>
```

Chapter that references a ditamap of content:

```xml
<chapter href="intro.ditamap" format="ditamap"/>
```

3.2.1.15 <colophon>
The <colophon> element references a topic describing how this document was created.

Usage information

In publishing, a colophon describes details of the production of a book. This information generally includes the typefaces used, and often the names of their designers; the paper, ink and details of the binding materials and methods might also receive mention. In the case of technical books, a colophon might specify the software used to prepare the text and diagrams for publication. The <colophon> can appear in the <frontmatter> or <backmatter>.

Specialization hierarchy
- map/topicref bookmap/colophon
Attributes

STUB CONTENT

Example

```xml
<bookmap>
  <title>Sample book</title>
  <!-- ... -->
  <backmatter>
    <colophon href="ProdNot.dita" navtitle="Production Notes"/>
  </backmatter>
</bookmap>
```

3.2.1.1.16 `<dedication>`

The `<dedication>` element references a topic containing a dedication for the book, such as to a person or group.

Specialization hierarchy

- map/topicref bookmap/dedication

Attributes

STUB CONTENT

Example

```xml
<frontmatter>
  <dedication href="dtm.dita"
    navtitle="Dedicated to Mother"/>
</frontmatter>
```

3.2.1.1.17 `<draftintro>`

The `<draftintro>` element references a topic used as an introduction to the draft of this book.

Specialization hierarchy

- map/topicref bookmap/draftintro

Attributes

STUB CONTENT

Example

```xml
<frontmatter>
  <draftintro href="introducing.dita"
    navtitle="Introduction to this draft"/>
</frontmatter>
```
3.2.1.1.18 <figurelist>
The <figurelist> element references a topic containing a list of figures in the book.

Usage information
It indicates to the processing software that the author wants a list of figures generated at the particular location. If no @href attribute is specified on the <figurelist> element, an external processor might generate a list of figures at this location.

Specialization hierarchy
- map/topicref bookmap/figurelist

Attributes
STUB CONTENT

Example
See the example in bookmap (37).

3.2.1.1.19 <frontmatter>
The <frontmatter> element contains the material that precedes the main body of a document.

Usage information
It might include items such as an abstract, a preface, and various types of book lists such as a <toc>, <tablelist>, or <figurelist>.

Specialization hierarchy
- map/topicref bookmap/frontmatter

Attributes
STUB CONTENT

Example
See the example in bookmap (37).

3.2.1.20 <glossarylist>
The <glossarylist> element references a list of glossary entries within the book. It indicates to the processing software that the author wants a glossary list generated at the particular location.

Specialization hierarchy
- map/topicref bookmap/glossarylist

Attributes
STUB CONTENT
Example
See backmatter (40).

3.2.1.1.21 <indexlist>
The <indexlist> element indicates to the processing software that the author wants an index at the particular location.

Processing expectations
If no @href attribute is specified on the <indexlist> element, an external processor might generate an index at this location. If the @href attribute references a topic or map, that topic or map contains a manually created index.

Specialization hierarchy
- map/topicref bookmap/indexlist

Attributes
STUB CONTENT

Example
See the example in bookmap (37).

3.2.1.1.22 <mainbooktitle>
The <mainbooktitle> element contains the primary title information for a book.

Specialization hierarchy
- topic/ph bookmap/mainbooktitle

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
See the example in bookmap (37).

3.2.1.1.23 <notices>
The <notices> element references a topic containing special notice information, for example, legal notices about supplementary copyrights and trademarks associated with the book.

Specialization hierarchy
- map/topicref bookmap/notices

Attributes
STUB CONTENT
Example
This example references a notices topic that contains legal content.

```xml
<backmatter>
  <notices href="notices.dita" navtitle="Legal notices"/>
</backmatter>
```

3.2.1.1.24 <part>
The <part> element references a part topic or a map that references part topics for the book.

Usage information
Use <part> to divide a document's chapters into logical groupings. For example, in a document that contains both guide and reference information, you can define two parts, one containing the guide information and the other containing the reference information.

Specialization hierarchy
- map/topicref bookmap/part

Attributes
STUB CONTENT

Example
Part topics that include chapters and subtopics:

```xml
<part href="guide.dita">
  <chapter href="intro.dita"/>
  <topicref href="caring.dita"/>
  <topicref href="feeding.dita"/>
</chapter>
<chapter href="setup.dita">
  <topicref href="prereq.dita"/>
  <topicref href="download.dita"/>
</chapter>
</part>

<part href="ref.dita">
  <chapter href="commands.dita">
    <topicref href="care.dita"/>
    <topicref href="feed.dita"/>
  </chapter>
  <chapter href="apis.dita">
    <topicref href="acare.dita"/>
    <topicref href="afeed.dita"/>
  </chapter>
</part>
```

Parts that reference ditamaps of content:

```xml
<part href="intro.ditamap" format="ditamap"/>
<part href="guide.ditamap" format="ditamap"/>
```
3.2.1.1.25 <preface>
The <preface> element references a topic or map containing introductory information about a book, such as the purpose and structure of the document.

Specialization hierarchy
- map/topicref bookmap/preface

Attributes
STUB CONTENT

Example
See the example in bookmap (37).

3.2.1.1.26 <tablelist>
The <tablelist> element references a topic that contains a list of tables within the book.

Processing expectations
This element indicates to the processing software that the author wants a list of tables generated at the particular location. If no @href attribute is specified on the <tablelist> element, an external processor might generate a list of tables at this location.

Specialization hierarchy
- map/topicref bookmap/tablelist

Attributes
STUB CONTENT

Example
See the example in bookmap (37).

3.2.1.1.27 <toc>
The <toc> element indicates to the processing software that the author wants a table of contents generated at the particular location.

Processing expectations
If no @href attribute is specified on the <toc> element, an external processor might generate a table of contents at this location. If the @href attribute references a topic or map, that topic or map contains a manually created table of contents.

Specialization hierarchy
- map/topicref bookmap/toc
Attributes
STUB CONTENT

Example
See the example in bookmap (37).

3.2.1.1.28 <trademarklist>
The <trademarklist> element references a topic that contains a list of trademarks within the book.

Processing expectations
This element indicates to the processing software that the author wants a list of trademarks generated at the particular location. If no @href attribute is specified on the <trademarklist> element, an external processor might generate a list of trademarks at this location.

Specialization hierarchy
- map/topicref bookmap/trademarklist

Attributes
STUB CONTENT

Example
See the example in bookmap (37).

3.2.1.2 Bookmap metadata elements
The Bookmap specialization of ditamap supports standard book production for collections of DITA topics. This section contains the metadata elements used by bookmap to store book-related metadata.

Note The OASIS document type for the bookmap specialization includes the xNAL domain specialization for describing the author or authors of a document. All elements that appear in that domain (the <authorinformation> element and all of its descendants) are described in the xNAL domain section, rather than in this bookmap metadata section.

3.2.1.2.1 <bookmeta>
The <bookmeta> element contains information about the book that is not considered book content, such as copyright information, author information, and any classifications.

Specialization hierarchy
- map/topicmeta bookmap/bookmeta

Attributes
STUB CONTENT

Example
In this example:
• The `<authorinformation>` contains the content for a reader's comment form; the `<maintainer>` element references that author information to create the reader comment form.

• The `<prodinfo>` element contains the book's product information. This includes the product name, and the version, release, and modification information.

• The `<bookid>` element contains the book's form number and part number information. The `<maintainer>` element contains information about the current maintainer of the book; it might reference the `<authorinformation>` element to provide detailed information about an author that is also the current maintainer.

• The `<bookrights>` element contains the book's copyright information.

This example makes heavy use of `<authorinformation>`, which is a domain element that OASIS delivers only with the `<bookmap>` specialization. However, other DTD, XSD Schema, or RELAX NG implementations of `<bookmap>` may not include the domain with `<authorinformation>`; likewise, `<authorinformation>` may be included in other document types.

```
<bookmeta>
  <authorinformation id="rcf">
    <organizationinfo>
      <namedetails>
        <organizationnamedetails>
          <organizationname>IBM</organizationname>
        </organizationnamedetails>
        <namedetails>ATTN: Dept XYZ<thoroughfare>3905 37th Street NW</thoroughfare>
          <locality>Rochester, MN<postalcode>55901</postalcode></locality>
          <country>USA</country>
        </addressdetails>
        <contactnumbers>
          <contactnumber type="telephone">800-555-1212</contactnumber>
          <contactnumber type="fax">800-555-1213</contactnumber>
        </contactnumbers>
        <emailaddresses><emailaddress>fred@example.com</emailaddress></emailaddresses>
        <urls><url>http://www.example.com/fred</url></urls>
      </organizationinfo>
      <prodinfo>
        <prodname>My Product</prodname>
        <vrmlist><vrm release="Release 1" version="Version 3"/></vrmlist>
      </prodinfo>
      <bookid>
        <bookpartno>99F9999</bookpartno>
        <booknumber>SC00-0000-00</booknumber>
        <maintainer href="#rcf"></maintainer>
      </bookid>
      <bookrights>
        <copyrfirst><year>1996</year></copyrfirst>
        <copyrlast><year>2006</year></copyrlast>
        <bookowner><organization>OASIS</organization></bookowner>
      </bookrights>
    </authorinformation>
  </organizationinfo>
</bookmeta>
```

### 3.2.1.2.2 `<approved>`

The `<approved>` element contains information about when and by whom the book was approved during its publication history.

#### Specialization hierarchy

- topic/data bookmap/approved

#### Attributes

**STUB CONTENT**
Example
See the example in bookchangehistory (52).

3.2.1.2.3 <bookchangehistory>
The <bookchangehistory> element contains information about a book's publishing lifecycle

Usage information
This element contains information about the history of the book's creation and publishing lifecycle, including who wrote, reviewed, edited, and tested the book. It also specifies when these events took place.

Specialization hierarchy
- topic/data bookmap/bookchangehistory

Attributes
STUB CONTENT

Example

```
<bookmeta>
  <bookchangehistory>
    <started><year>2007</year><month>10</month></started>
    <completed><year>2008</year><month>01</month></completed>
    </reviewed>
    <edited>
      <person>Joe T. Editor</person>
      <completed><year>2008</year><month>03</month><day>15</day></completed>
    </edited>
    <tested>
      <organization>OASIS</organization>
      <completed><year>2008</year><month>04</month></completed>
    </tested>
    <approved>
      <organization>OASIS</organization>
      <completed><year>2008</year><month>05</month></completed>
    </approved>
    <bookevent>
      <bookeventtype name="Type of Major Event"/>
      <completed><year>2009</year></completed>
    </bookevent>
  </bookchangehistory>
</bookmeta>
```

3.2.1.2.4 <bookevent>
The <bookevent> element indicates a general event in the publication history of a book.

Usage information
This element is appropriate for specialization if the current set of specific book event types does not meet your needs. If an element already exists to describe a specific type of event, such as <reviewed>, <edited>, or <approved>, use that element instead.
Specialization hierarchy
- topic/data bookmap/bookevent

Attributes
STUB CONTENT

Example
See the example in bookchangehistory (52).

3.2.1.2.5 <bookeventtype>
The <bookeventtype> element indicates the specific nature of a <bookevent>, such as updated, indexed, or deprecated.

Usage information
The required @name attribute indicates the event's type.

Specialization hierarchy
- topic/data bookmap/bookeventtype

Attributes
STUB CONTENT

Example
See the example in bookchangehistory (52).

3.2.1.2.6 <bookid>
The <bookid> element contains the publisher's identification information for the book, such as part number, edition number and ISBN number.

Specialization hierarchy
- topic/data bookmap/bookid

Attributes
STUB CONTENT

Example
See bookmeta (50).

3.2.1.2.7 <booknumber>
The <booknumber> element contains the book's form number, such as SC21-1920.
Attributes
STUB CONTENT

Example
In this example, "99F1234" is a part number assigned to this book by the publisher, while SC21-1234-00 is a number that identifies this book among all of the author's works.

```
<bookmeta>
  <bookid>
    <bookpartno>99F1234</bookpartno>
    <booknumber>SC21-1234-00</booknumber>
  </bookid>
</bookmeta>
```

3.2.1.2.8 <bookowner>
The <bookowner> element specifies the owner of the book's copyright.

Specialization hierarchy
- topic/data bookmap/bookowner

Attributes
STUB CONTENT

Example
```
<bookmeta>
  <bookrights>
    <copyrfirst><year>1994</year></copyrfirst>
    <copyrlast><year>2006</year></copyrlast>
    <bookowner>
      <organization>Example Corporation</organization>
    </bookowner>
  </bookrights>
</bookmeta>
```

3.2.1.2.9 <bookpartno>
The <bookpartno> element contains the book's part number, such as 99F1234.

Usage information
A publisher might use a number like this one to identify a book for tracking purposes.

Specialization hierarchy
- topic/data bookmap/bookpartno

Attributes
STUB CONTENT
Example
In this example, "99F1234" is a part number assigned to this book by the publisher, while SC21-1234-00 is a number that identifies this book among all of the author's works.

```xml
<bookmeta>
  <bookid>
    <bookpartno>99F1234</bookpartno>
    <booknumber>SC21-1234-00</booknumber>
  </bookid>
</bookmeta>
```

3.2.1.2.10 <bookrestriction>
The <bookrestriction> element indicates whether the book is classified or restricted in some way.

**Specialization hierarchy**
- topic/data bookmap/bookrestriction

**Attributes**
The @value attribute indicates any restrictions on the use of the material, such as declaring the information confidential or for licensed use only.

STUB CONTENT

Example

```xml
<bookrights>
  <copyrfirst><year>1994</year></copyrfirst>
  <copyrlast><year>2006</year></copyrlast>
  <bookowner><organization>Example Corporation</organization></bookowner>
  <bookrestriction value="unclassified"/>
</bookrights>
```

3.2.1.2.11 <bookrights>
The <bookrights> element contains the information about the legal rights associated with the book, including copyright dates and owners.

**Specialization hierarchy**
- topic/data bookmap/bookrights

**Attributes**
STUB CONTENT

Example
See bookmeta (50).
3.2.1.2.12 <completed>
The <completed> element indicates a completion date for some type of book event, such as a review, editing, or testing.

Specialization hierarchy
- topic/ph bookmap/completed

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
See the example in bookchangehistory (52).

3.2.1.2.13 <copyrfirst>
The <copyrfirst> element contains the copyright year, or the first copyright year within a multiyear copyright statement.

Specialization hierarchy
- topic/data bookmap/copyrfirst

Attributes
STUB CONTENT

Example
See the example in bookmap (37).

3.2.1.2.14 <copyrlast>
The <copyrlast> element contains the last copyright year within a multiyear copyright statement.

Specialization hierarchy
- topic/data bookmap/copyrlast

Attributes
STUB CONTENT

Example
See the example in bookmap (37).
3.2.1.2.15 <day>
The <day> element denotes a day of the month.

**Specialization hierarchy**
- topic/ph bookmap/day

**Attributes**
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

**Example**
```
<bookchangehistory>
  <edited>
    <person>Joe T. Editor</person>
    <completed><year>2008</year><month>10</month><day>13</day></completed>
  </edited>
</bookchangehistory>
```

3.2.1.2.16 <edited>
The <edited> element contains information about when and by whom the book was edited during its publication history.

**Specialization hierarchy**
- topic/data bookmap/edited

**Attributes**
STUB CONTENT

**Example**
See the example in bookchangehistory (52).

3.2.1.2.17 <edition>
The <edition> element contains the edition number information, such as First Edition, or Third Edition, used by a publisher to identify a book.

**Specialization hierarchy**
- topic/data bookmap/edition

**Attributes**
STUB CONTENT

**Example**
```
<bookmeta>
  <bookid>
    <edition>1</edition>
  </bookid>
</bookmeta>
```
3.2.1.2.18 <isbn>

Specialization hierarchy
- topic/data bookmap/isbn

Attributes
STUB CONTENT

Example
<bookmeta>
  <bookid>
    <isbn>978-0141000039</isbn>
  </bookid>
</bookmeta>

3.2.1.2.19 <maintainer>
The <maintainer> element contains information about who maintains the document; the maintainer might be an organization or a person.

Specialization hierarchy
- topic/data bookmap/maintainer

Attributes
STUB CONTENT

Example
See the example in bookmeta (50).

3.2.1.2.20 <month>
The <month> element denotes a month of the year.

Specialization hierarchy
- topic/ph bookmap/month

Attributes
The following attributes are available on this element: **Universal attribute group** (159) and @keyref (180).

Example
<bookchangehistory>
  <edited>
    <person>Joe T. Editor</person>
  </edited>
</bookchangehistory>
3.2.1.2.21 <organization>
The <organization> element contains the name of an organization.

Usage information
Note that unlike <organizationname>, the <organization> element is not restricted to use within <authorinformation>; it does not have to contain the name of an authoring organization.

Specialization hierarchy
- topic/data bookmap/organization

Attributes
STUB CONTENT

Example

```xml
<bookmeta>
  <bookrights>
    <copyrfirst><year>1996</year></copyrfirst>
    <copyrlast><year>2006</year></copyrlast>
    <bookowner><organization>OASIS</organization></bookowner>
  </bookrights>
</bookmeta>
```

3.2.1.2.22 <person>
The <person> element contains information about the name of a person.

Usage information
Note that unlike the <personname> element, the <person> element is not restricted to describing the names of authors.

Specialization hierarchy
- topic/data bookmap/person

Attributes
STUB CONTENT

Example

```xml
<bookmeta>
  <bookrights>
    <copyrfirst><year>1977</year></copyrfirst>
    <copyrlast><year>2008</year></copyrlast>
    <bookowner><person>Jane Doe</person></bookowner>
  </bookrights>
</bookmeta>
```
3.2.1.2.23 <printlocation>
The <printlocation> element indicates the location where the book was printed.

Usage information
Customarily, the content is restricted to the name of the country.

Specialization hierarchy
- topic/data bookmap/printlocation

Attributes
STUB CONTENT

Example
See the example in publisherinformation (60).

3.2.1.2.24 <published>
The <published> element contains information about the person or organization publishing the book, the dates when it was started and completed, and any special restrictions associated with it.

Specialization hierarchy
- topic/data bookmap/published

Attributes
STUB CONTENT

Example
See the example in publisherinformation (60).

3.2.1.2.25 <publisherinformation>
The <publisherinformation> contains information about what group or person published the book, where it was published, and certain details about its publication history.

Usage information
Other publication history information is found in the <bookchangehistory> element.

Specialization hierarchy
- topic/publisher bookmap/publisherinformation

Attributes
STUB CONTENT
**Example**

```xml
<bookmeta>
  <publisherinformation>
    <organization>Example Publishers</organization>
    <printlocation>Austin, TX</printlocation>
    <published>
      <publishtype value="general"/>
      <completed><year>1977</year></completed>
    </published>
  </publisherinformation>
</bookmeta>
```

### 3.2.1.2.26 <publishtype>

The `<publishtype>` element indicates whether the book is generally available from the publisher or is restricted in some way.

**Usage information**

The `@value` attribute indicates the restrictions, such as beta release, limited availability, or general availability.

**Specialization hierarchy**

- topic/data bookmap/publishtype

**Attributes**

STUB CONTENT

**Example**

See the example in `publisherinformation` (60).

### 3.2.1.2.27 <reviewed>

The `<reviewed>` element contains information about when and by whom the book was reviewed during its publication history.

**Specialization hierarchy**

- topic/data bookmap/reviewed

**Attributes**

STUB CONTENT

**Example**

See the example in `bookchangehistory` (52).
3.2.1.2.28 <revisionid>
The <revisionid> element indicates the revision number or revision ID of the book.

Usage information
The processing implementation determines how the level is displayed. Common methods include using a dash, for example “-01”, or a period, such as “.01”.

Specialization hierarchy
- topic/ph bookmap/revisionid

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example

```xml
<bookchangehistory>
  <edited>
    <person>Joe T. Editor</person>
    <revisionid>1</revisionid>
    <completed><year>2008</year><month>03</month><day>15</day></completed>
  </edited>
  <edited>
    <person>Joe T. Editor</person>
    <revisionid>2</revisionid>
    <completed><year>2008</year><month>10</month><day>13</day></completed>
  </edited>
</bookchangehistory>
```

3.2.1.2.29 <started>
The <started> element indicates a start date for some type of book event, such as a review, editing, or testing.

Specialization hierarchy
- topic/ph bookmap/started

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
See the example in bookchangehistory (52).
3.2.1.2.30 <summary>
The <summary> element contains a text summary associated with a book event (such as <approved> or <reviewed>) or with the list of copyrights for the book.

Specialization hierarchy
- topic/ph bookmap/summary

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example

```xml
<bookchangehistory>
  <edited>
    <person>Joe T. Editor</person>
    <revisionid>1</revisionid>
    <completed><year>2008</year><month>03</month><day>15</day></completed>
    <summary>Added several new topics</summary>
  </edited>
  <edited>
    <person>Joe T. Editor</person>
    <revisionid>2</revisionid>
    <completed><year>2008</year><month>10</month><day>13</day></completed>
    <summary>Fixed a few typos</summary>
  </edited>
</bookchangehistory>
```

3.2.1.2.31 <tested>
The <tested> element contains information about when and by whom the book was tested during its publication history.

Specialization hierarchy
- topic/data bookmap/tested

Attributes
STUB CONTENT

Example
See the example in bookchangehistory (52).

3.2.1.2.32 <volume>
The <volume> element contains the book's volume number, such as "2" to represent Volume 2.

Specialization hierarchy
- topic/data bookmap/volume

Attributes
STUB CONTENT
3.2.1.2.33 `<year>`

The `<year>` element denotes a year.

**Specialization hierarchy**

- topic/ph bookmap/year

**Attributes**

The following attributes are available on this element: Universal attribute group (159) and `@keyref` (180).

**Example**

```xml
<bookchangehistory>
  <edited>
    <person>Joe T. Editor</person>
    <completed>
      <year>2008</year>
      <month>10</month>
      <day>13</day>
    </completed>
  </edited>
</bookchangehistory>
```

3.2.2 Concept elements

DITA concept topics answer "What is..." questions. Use the concept topic to introduce the background or overview information for tasks or reference topics. The concept topic restricts content following a section or example to other sections or examples. For more details on when to use concept and other information types, please refer to the DITA architectural specification.

3.2.2.1 `<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, but generally the structure of a concept is fairly simple.

**Specialization hierarchy**

- topic/topic concept/concept

**Attributes**

STUB CONTENT
3.2.2.2 <conbody>
The <conbody> element is the main body-level element for a concept.

Usage information
Like the <body> element of a general <topic>, <conbody> allows paragraphs, lists, and other elements as well as sections and examples. However, <conbody> 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
- topic/body concept/conbody

Attributes
STUB CONTENT

Example
See the example in concept (64).

3.2.2.3 <conbodydiv>
The <conbodydiv> element is similar to the <bodydiv> element in that it provides an informal container for content that might be grouped within a concept.

Usage information
There are no additional semantics attached to the <conbodydiv> element; it is purely a grouping element provided to help organize content.

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.

Specialization hierarchy
- topic/bodydiv concept/conbodydiv

Attributes
The following attributes are available on this element: Universal attribute group (159).
Example

One common use case for the `<conbodydiv>` element is to group a sequence of sections for reuse, so that another concept can reference the entire set with a single `@conref` attribute.

```xml
<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 might be reused elsewhere.</p>
    <conbodydiv id="my_conbodydiv">
      <section><title>First</title> ... </section>
      <section><title>Second</title> ... </section>
    </conbodydiv>
  </conbody>
</concept>
```

3.2.3 Glossary elements

Glossary elements include those elements designed to specify terms and their definitions, as well as elements that are designed to group, reference, or otherwise make use of information in the glossentry topic.

3.2.3.1 Glossentry elements

Use the glossary entry topic type to define glossary terms. Each glossentry topic should define a single sense of a term.

3.2.3.1.1 `<glossentry>`

The `<glossentry>` element defines a single sense of a glossary term. Glossary entries for different term senses can be reused independently of one another.

Processing expectations

The recommended (but not required) book processing is to sort and group glossary entries based on the localized term so a back-of-the-book glossary can contain a collated list of terms with the definitions of the individual senses of each term indented under the term. The glossary can have a different organization in different languages depending on the translation of the terms. One possible implementation of a glossary in online processing is to associate a hotspot for mentions of terms in `<term>` elements and display the definition on hover or click.

Specialization hierarchy

- topic/topic concept/concept glossentry/glossentry

Attributes

STUB CONTENT

Example

Figure 5: The glossary term "USB flash drive" with additional information

```xml
<glossentry id="usbfd">
  <glossterm>USB flash drive</glossterm>
  <glossdef>A small portable drive.</glossdef>
  <glossBody>
    <glossPartOfSpeech value="noun"/>
  </glossBody>
</glossentry>
```
3.2.3.1.2 <glossterm>
The <glossterm> element specifies the preferred term that is associated with a definition of a sense. If the same term has multiple senses, create a separate <glossentry> topic for each sense.

Specialization hierarchy
- topic/title concept/title glossentry/glossterm

Attributes
STUB CONTENT

Example
See the example in <glossentry>. (66)

3.2.3.1.3 <glossdef>
The <glossdef> element specifies the definition of one sense of a term. If a term has multiple senses, create a separate <glossentry> topic to define each sense.

Specialization hierarchy
- topic/abstract concept/abstract glossentry/glossdef

Attributes
STUB CONTENT

Example
See the example in <glossentry>. (66)
3.2.3.1.4 <glossAbbreviation>
The <glossAbbreviation> element provides an abbreviated form of the term that is contained in a <glossterm> element.

Specialization hierarchy
- topic/title concept/title glossentry/glossAbbreviation

Attributes
STUB CONTENT

Example
See the example in <glossentry>. (66)

3.2.3.1.5 <glossAcronym>
The <glossAcronym> element defines an acronym for the term that is defined in the <glossterm> element.

Usage information
This element can be used with the <abbreviated-form> element to display an expanded version of an acronym the first time that acronym appears in a set of text. See <abbreviated-form> (75) for information on how the two elements interact.

Note Several issues arise when acronyms are translated into other languages. For example, an acronym in one language might not have an equivalent in another language. When acronyms are first displayed, some languages will display the expanded form first followed by the acronym in parenthesis, while other languages do the reverse. For some acronyms, a translation might need to render both the original and the translated version of the acronym. For all of these reasons, DITA allows an author or translator to control what is presented to a reader by using the <glossSurfaceForm> element, which will often accompany the <glossAcronym>. The <abbreviated-form> (75) topic contains information on how the <glossSurfaceForm> and <glossAcronym> elements affect references to the primary term.

Specialization hierarchy
- topic/title concept/title glossentry/glossAcronym

Attributes
STUB CONTENT

Example
See the example in <glossentry>. (66)

3.2.3.1.6 <glossAlt>
The <glossAlt> element contains a variant term for the preferred term. The variant should have the same meaning as the term in the <glossterm> element; the variant is simply another way to refer to the same term. There might be many ways to refer to a term; each variant is placed in its own <glossAlt>
The `<glossUsage>` element can be used within `<glossAlt>` to indicate when use of the alternate term is appropriate.

**Usage information**

- **Note** A list of alternative terms is specific to the language, so the translation of a `<glossentry>` topic might result in empty elements within a `<glossAlt>` container.

**Specialization hierarchy**

- topic/section concept/section glossentry/glossAlt

**Attributes**

The following attributes are available on this element: [Universal attribute group](159).

**Example**

See the example in `<glossentry>`.

### 3.2.3.1.7 `<glossAlternateFor>`

The `<glossAlternateFor>` element indicates when a variant term has a relationship to another variant term in addition to the preferred term.

**Usage information**

The `<glossAlternateFor>` element is available inside the `<glossAlt>` element, which is a container that provides a variant for the primary `<glossentry>` term. In some cases, the variant might also be an alternate for another term. In the example below, the abbreviation "stick" is a variant of the primary term (USB flash drive). The `<glossAlternateFor>` element indicates that "stick" is also a variant of the synonym "memory stick".

**Specialization hierarchy**

- topic/xref concept/xref glossentry/glossAlternateFor

**Attributes**

The following attributes are available on this element: [STUB CONTENT](24), [STUB CONTENT](24) (with a narrowed definition of `@href`, given below), [keyref](24), and #unique_24/unique_24_Connect_42_outputclass.

- **@href**

  References a term for which the current variant is an alternate (in addition to the primary term of this `<glossentry>` topic). The reference will often be to another `<glossAlt>` element within the same `<glossentry>` topic, indicating that the current variant is an alternate for both the primary term and the referenced alternate term. See [STUB CONTENT](24) for details on syntax.

**Example**

See the example in `<glossentry>`.
3.2.3.1.8 <glossBody>
The <glossBody> element is used to provide details about a glossary term (such as part of speech or additional forms of the term).

Usage information

Formatting expectations

Processing expectations

Specialization hierarchy
- topic/body concept/conbody glossentry/glossBody

Attributes
STUB CONTENT

Example
See the example in <glossentry>. (66)

3.2.3.1.9 <glossPartOfSpeech>
The <glossPartOfSpeech> element identifies the part of speech for the preferred and alternate terms. Alternate terms must have the same part of speech as the preferred term, because all terms in the <glossentry> topic designate the same subject.

Usage information
If validation is required for parts of speech, use a subject scheme to set up controlled values for the @value attribute. Alternatively, processing rules can be used to validate the values. This avoids the need to hard code values in the XML grammar files, which would limit extension and would make values difficult to change for different languages.

Specialization hierarchy
- topic/data concept/data glossentry/glossPartOfSpeech

Attributes
The following attributes are available on this element: STUB CONTENT (with a narrowed definition of @value, given below), STUB CONTENT, STUB CONTENT, and #unique_24/unique_24_Connect_42_outputclass.

@value
   The part of speech for the term or terms.

Example
See the example in <glossentry>. (66)
3.2.3.1.10 <glossProperty>
The <glossProperty> element is a specialization of the <data> element. You can use it to specify additional details about the preferred term or its subject, for example, the gender of a noun.

Specialization hierarchy
- topic/data concept/data glossentry/glossProperty

Attributes
STUB CONTENT

Example

```xml
<glossentry id="algorithm" xml:lang="es-es">
  <glossterm>El algoritmo</glossterm>
  <glossdef>Un algoritmo define un método de calcular un resultado.</glossdef>
  <glossBody>
    <glossPartOfSpeech value="noun"/>
    <glossProperty name="gender" value="masculine"/>
  </glossBody>
</glossentry>
```

3.2.3.1.11 <glossScopeNote>
The <glossScopeNote> element contains a clarification of the subject that is designated by the <glossterm>, such as examples of included or excluded companies or products. For instance, a scope note for "Linux" might explain that the term does not apply to UNIX products; it might give examples of Linux products that are included as well as UNIX products that are excluded.

Usage information

Specialization hierarchy
- topic/note concept/note glossentry/glossScopeNote

Attributes
STUB CONTENT

Example

```xml
<glossentry id="linuxOS" xml:lang="en-us">
  <glossterm>Linux Operating System</glossterm>
  <glossdef>An operating system based on the kernel created by Linus Torvald.</glossdef>
  <glossBody>
    <glossPartOfSpeech value="noun"/>
    <glossScopeNote>Doesn't apply to UNIX products that bundle other kernels. Also, doesn't apply to the Linux Open Source Project that work on Linux distributions but, instead, only to the distributions themselves. Examples include RedHat, SuSE, and Ubuntu. </glossScopeNote>
  </glossBody>
</glossentry>
```
3.2.3.12 <glossShortForm>
The <glossShortForm> element provides a shorter alternative to the primary term that is specified in the <glossterm> element.

Usage information

Note Any list of alternative terms is, of course, specific to the language, so translation of a <glossentry> topic might result in an empty <glossShortForm> element if there is no equivalent in the target language.

Specialization hierarchy
- topic/title concept/title glossentry/glossShortForm

Attributes
STUB CONTENT

Example

```xml
<glossentry id="www">
  <glossterm>World Wide Web</glossterm>
  <glossdef>A collection of documents available through the Internet.</glossdef>
  <glossBody>
    <glossPartOfSpeech value="noun"/>
    <glossAlt>
      <glossShortForm>the Web</glossShortForm>
    </glossAlt>
  </glossBody>
</glossentry>
```

3.2.3.13 <glossStatus>
The <glossStatus> element identifies the usage status of a preferred or alternate term. If the status isn't specified, the <glossterm> provides a preferred term and an alternate term provides an allowed term.

Usage information

If validation is required for status values, use a subject scheme to set up controlled values for the @value attribute. Alternatively, processing rules can be used to validate the values.

Specialization hierarchy
- topic/data concept/data glossentry/glossStatus

Attributes

The following attributes are available on this element: STUB CONTENT (with a narrowed definition of @value, given below), STUB CONTENT, STUB CONTENT, and #unique_24/unique_24_Connect_42_outputclass.

@value
The usage status of the term or alternate term.
Example
See the example in <glossentry>. (66)

3.2.3.1.14 <glossSurfaceForm>
The <glossSurfaceForm> element specifies an unambiguous presentation of the <glossTerm> that might combine multiple forms. The surface form is suitable to introduce the term in new contexts.

Usage information
The <glossSurfaceForm> element is most often used for terms that also specify the <glossAcronym> element. In that context, the <glossSurfaceForm> element contains the term in a manner that introduces both the term and the acronym, so that later references to the term can be replaced with the acronym alone. See the <abbreviated-form> (75) element for a full description of how the surface form is used together with acronyms.

Specialization hierarchy
- topic/p concept/p glossentry/glossSurfaceForm

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
The following topic defines the term "Anti-lock Braking System". Within the topic, the <glossSurfaceForm> element provides a version of the term that combines both the primary term and the acronym. This allows references to the topic from <abbreviated-form> to different versions of the term in different contexts.

```xml
<glossentry id="abs">
  <glossTerm>Anti-lock Braking System</glossTerm>
  <glossBody>
    <glossSurfaceForm>Anti-lock Braking System (ABS)</glossSurfaceForm>
    <glossAlt>
      <glossAcronym>ABS</glossAcronym>
    </glossAlt>
  </glossBody>
</glossentry>
```

3.2.3.1.15 <glossSymbol>
The <glossSymbol> element identifies a standard image that is associated with the subject of the <glossterm> element.

Specialization hierarchy
- topic/image concept/image glossentry/glossSymbol

Attributes
The following attributes are available on this element: STUB CONTENT, #unique_24/unique_24_Connect_42_outputclass, @keyref, and the attributes defined below.

STUB CONTENT
STUB CONTENT
**Example**

```xml
<glossentry id="atlanticpuffin">
  <glossterm>Atlantic Puffin</glossterm>
  <glossdef>A sea bird that lives in the Atlantic</glossdef>
  <glossBody>
    <glossSymbol href="puffinicon.jpg" scope="local">
      <alt>Atlantic puffin icon</alt>
    </glossSymbol>
  </glossBody>
</glossentry>
```

### 3.2.3.1.16 `<glossSynonym>`

The `<glossSynonym>` element provides a term that is a synonym of the primary value in the `<glossterm>` element.

**Usage information**

- **Note** A list of synonyms is specific to the language, so translation of a `<glossentry>` topic might result in an empty `<glossSynonym>` element.

**Specialization hierarchy**

- topic/title concept/title glossentry/glossSynonym

**Attributes**

STUB CONTENT

**Example**

See the example in `<glossentry>`.(66)
3.2.3.17 <glossUsage>
The <glossUsage> element provides information about the correct use of a term, such as where or how it can be used.

Specialization hierarchy
- topic/note concept/note glossentry/glossUsage

Attributes
STUB CONTENT

Example
See the example in <glossentry>. (66)

3.2.3.2 Glossary related elements
Elements related to the glossary specialization are not defined as part of the glossary topic type, but are often used in conjunction with those elements.

3.2.3.2.1 <abbreviated-form>
The <abbreviated-form> element represents a reference to a term that might appear in an abbreviated form (often an acronym).

Usage information
The long and short forms of the term are generally defined in a <glossentry> topic. Processors display the referenced term when rendering an <abbreviated-form> element.

Formatting expectations
The <abbreviated-form> element is intended to reference a <glossentry> topic that contains both a term and an abbreviated form of that term. The topic might also provide a surface form that differs from the original term. The full term or surface form is rendered in introductory contexts where the term might be unfamiliar to a reader or in other contexts where a precise term is appropriate. In other contexts a processor substitutes the abbreviated form of the term. Note that the definition of an introductory context will differ for every deliverable format.

For instance, a process composing a book deliverable might emit the surface form of a term on the first reference to the <glossentry> topic within the book or for every reference within a copyright or a warranty-related warning. A process generating an online page might emit the surface form as a hover tooltip on every instance of the term.

The following rules determine how to render an <abbreviated-form> element. As described, the definition of “introductory context” will vary based on the rendered format and processor.

1. If the referenced topic is not a <glossentry> topic (or a specialization of <glossentry>), the title of the topic SHOULD be displayed.
2. Otherwise, if <abbreviated-form> is located in an introductory context, and the referenced topic contains a non-empty <glossSurfaceForm> element, processors SHOULD render the contents of the <glossSurfaceForm> element from the referenced <glossentry> topic.
3. Otherwise, if <abbreviated-form> is located in an introductory context, processors SHOULD render the contents of the <glossTerm> element from the referenced <glossentry> topic.
4. Otherwise (in non-introductory contexts), if the referenced <glossentry> topic contains a non-empty <glossAcronym> element, processors SHOULD render the abbreviated form of the term by displaying the contents of the <glossAcronym> element from the referenced <glossentry> topic.

5. Otherwise, processors SHOULD render the contents of the <glossterm> element from the referenced <glossentry> topic.

For instance, if an <abbreviated-form> element with the attribute keyref="abs" provided the first occurrence of the ABS term within a book, the sentence could be rendered as follows:

“The Anti-lock Brake System (ABS) will prevent the car from skidding in adverse weather conditions.”

If the ABS term had appeared previously within the book, the same sentence could instead be rendered as follows:

“The ABS will prevent the car from skidding in adverse weather conditions.”

**Processing expectations**

**Specialization hierarchy**

+ topic/term abbrev-d/abbreviated-form

**Attributes**

The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

**Example**

The term and acronym can be defined as follows, in a <glossentry> topic. Note that the @id attribute of the topic does not need to match the term or acronym.

```xml
<glossentry id="abs-definition">
  <glossterm>Anti-lock Braking System</glossterm>
  <glossBody>
    <glossSurfaceForm>Anti-lock Braking System (ABS)</glossSurfaceForm>
    <glossAlt>
      <glossAcronym>ABS</glossAcronym>
    </glossAlt>
  </glossBody>
</glossentry>
```

Note that there are three important elements for the purposes of rendering the <abbreviated-form> element.

- The <glossSurfaceForm> element defines the term as it should be displayed in an introductory context.
- The <glossAcronym> element defines the acronym associated with this term.
- The <glossterm> element provides a fallback version of the term, which will be displayed in situations where the preferred representation is unavailable.

The <glossentry> topic will be added to a map in the following manner. Again, the key (in this case "abs") does not need to match the term or acronym value.

```xml
<glossref keys="abs" href="antilock.dita"/>
```
An author who wants to reference this topic can do so by using the `<abbreviated-form>` element. The @keyref attribute references the value defined on the @keys attribute above.

```xml
<section>An <abbreviated-form keyref="abs"/> helps a driver to stop. For this reason many find an <abbreviated-form keyref="abs"/> useful.
</section>
```

When rendered, the introductory usage of "abs" will display the surface form of the referenced term, while the later uses will display the acronym, as demonstrated here.

An Anti-lock Braking System (ABS) helps a driver to stop. For this reason many find an ABS useful.

### 3.2.3.2.2 `<glossref>`

The `<glossref>` element is a convenience element in maps for creating a reference to a glossary topic. It has a required @keys attribute, which forces the author to create a key by which inline terms can reference their definition. For example, when `<glossentry>` topics are used to define acronyms, this reminds authors to create a key which `<abbreviated-form>` elements can use to reference the short and expanded versions of that acronym.

#### Usage information

Note that the key value does not need to match the target term or acronym. In fact, using a more qualified value for the @keyref will reduce conflicts in situations where the same term or acronym might resolve in many ways. For example, an information set could use "cars.abs" as the key for the term Anti-lock Braking System, and "ship.abs" to refer to the term American Bureau of Shipping.

#### Specialization hierarchy

+ map/topicref glossref-d/glossref

#### Attributes

The following attributes are available on this element: STUB CONTENT, STUB CONTENT, STUB CONTENT (with a narrowed definition of @href, given below), #unique_24/unique_24_Connect_42_outputclass, @keyref, and the attributes defined below. This element also uses @processing-role, @collection-type, @locktitle, @chunk, and @search from STUB CONTENT; this element also uses narrowed definitions of @linking, @toc, and @print from STUB CONTENT, given below.

- **@href**
  A pointer to a glossary definition, typically a `<glossentry>` topic. See STUB CONTENT for detailed information on supported values and processing implications. STUB CONTENT

- **@keys** STUB CONTENT
  Associates one or more space-delimited keys with the target of the glossary reference. See STUB CONTENT for information on using the attribute.
@linking
Defines some specific linking characteristics of a glossary topic's current location in the map. On this element the value defaults to "none" in order to keep individual glossary entries from creating links based on their location in the map. Allowable values are:

- **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.

### Example

```xml
<map>
  <!-- ... -->
  <topicref href="car-maintenance.dita"/>
  <!-- ... -->
  <glossref keys="cars.abs" href="antiLockBrake.dita"/>
  <!-- ... key declarations for other referenced acronyms ... -->
</map>
```

### 3.2.3.2.3 <glossgroup>

The `<glossgroup>` is a specialized topic that can contain multiple `<glossentry>` topics within a single document.

#### Specialization hierarchy
- topic/topic concept/concept glossgroup/glossgroup

#### Attributes

STUB CONTENT

### Example

```xml
<glossgroup id="things" xml:lang="en">
  <title>Some terms</title>
  <glossentry id="bicycle">
    <glossterm>bicycle</glossterm>
    <glossdef>Human powered mode of transport with two wheels</glossdef>
  </glossentry>
  <glossentry id="fruitbat">
    <glossterm>Fruit bat</glossterm>
    <glossdef>A bat which likes fruit</glossdef>
  </glossentry>
</glossgroup>
```

### 3.2.4 Reference elements

Reference topics describe factual material about a subject, such as the commands in a programming language. This format is also suitable for bibliographies, catalogues, the list of ingredients for recipes, and
similar collections of structured descriptive prose. For more details on when to use reference and other information types, please refer to the DITA architectural specification.

3.2.4.1 <reference>
The <reference> element defines a top-level container for a reference topic. Reference topics document programming constructs or facts about a product. Examples of reference topics include (but are not limited to) product specifications, environmental specifications, equipment lists, parts lists, required tools, language elements, class descriptions, commands, functions, and API information. All of these topics provide quick access to facts, but they do not contain a deeper explanation of related concepts or tasks.

Usage information
Reference topics have the same high-level structure as any other topic type, with a title, short description, and body. Within the body, reference topics are typically organized into one or more sections, property lists, and tables. The reference topic type provides general rules that apply to all kinds of reference information, using elements like <refsyn> for syntax or signatures, and <properties> for lists of properties and values.

Specialization hierarchy
The <reference> element is specialized from the <topic> element in the topic module.

Attributes
STUB CONTENT

Example
Figure 6: Reference topic for software material

```xml
<reference id="refexample">
  <title>A reference topic</title>
  <refbody>
    <refsyn>Describe command or api syntax here, possibly using synph or syntax elements markup for explicit definition of syntax or prototype construction.</refsyn>
    <section><title>Some section title</title></section>
    <properties>
      <property>
        <proptype>type</proptype>
        <propvalue>value</propvalue>
        <propdesc>description</propdesc>
      </property>
    </properties>
  </refbody>
</reference>
```

Figure 7: Reference topic for hardware maintenance
The following information could apply to an entire set of maintenance procedures, each of which would link to this topic.

```xml
<reference id="requiredTools">
  <title>Tools required to maintain a big machine</title>
  <refbody>
    <section><title>Small tools</title></section>
    <ul>
      <li>Hard hat</li>
    </ul>
  </refbody>
</reference>
```
3.2.4.2 <refbody>
The <refbody> element is a container for the main content of the reference topic. Reference topics limit the body structure to tables (both simple and standard), property lists, syntax sections, and generic sections and examples, in any sequence or number.

Usage information
Reference topics represent the kind of information that users typically consult to understand programming objects, configuration file options, recipes, terminological descriptions, product or other specifications, equipment or parts lists, or any other set of factual information.

Specialization hierarchy
The <refbody> element is specialized from the <body> element in the topic module.

Attributes
STUB CONTENT

Example
See reference (79).

3.2.4.3 <refbodydiv>
The <refbodydiv> element is similar to the <bodydiv> element in that it provides an informal container for content that might be grouped within a reference. Reference topics place many restrictions on their content compared to generic topics; the <refbodydiv> element maintains these restrictions by only allowing elements that are already available within the body of a reference. There are no additional semantics attached to the <refbodydiv> element; it is purely a grouping element provided to help organize content.

Usage information
The <refbodydiv> element can nest itself, which means that it can be specialized to create structured information within a specialized reference topic. Another common use case for the <refbodydiv> element is to group a sequence of related elements for reuse, so that another topic can reference the entire set with a single @conref attribute.

Specialization hierarchy
The <refbodydiv> element is specialized from the <bodydiv> element in the topic module.
Attributes

The following attributes are available on this element: Universal attribute group (159).

Example

```xml
<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>
```

### 3.2.4.4 `<refsyn>`

The `<refsyn>` element is a special section inside a reference topic. The section often contains syntax or signature content (for example, the calling syntax for a command-line utility or an API signature). The `<refsyn>` contains a brief, possibly diagrammatic description of the subject's interface or high-level structure.

**Specialization hierarchy**

The `<refsyn>` element is specialized from the `<section>` element in the topic module.

**Attributes**

STUB CONTENT

**Example**

*Figure 8: Reference topic for software material*

```xml
<reference id="MyAPI">
  <title>MyAPI</title>
  <refbody>
    <refsyn>Describe the MyAPI syntax here, possibly using synph or syntax elements markup for explicit definition of syntax or prototype construction.</refsyn>
  </refbody>
</reference>
```

### 3.2.4.5 `<properties>`

The `<properties>` element gives a list of properties for the subject of the current topic, for example whether a class is public or protected. Each property can include the type, value, and a description.

**Usage information**

To represent multiple values for a single type, create additional property elements and use only the `<propvalue>` element (and `<propdesc>` when needed) for each successive value.
Formatting expectations
The typical rendering is usually in a table-like format.

Specialization hierarchy
The `<properties>` element is specialized from the `<simpletable>` element in the topic module.

Attributes
STUB CONTENT

Example
<properties>
  <prophead>
    <proptypehd>Visual Element</proptypehd>
    <propvaluehd>Value</propvaluehd>
    <propdeschd>Implication</propdeschd>
  </prophead>
  <property>
    <proptype>color</proptype>
    <propvalue>red</propvalue>
    <propdesc>depicts anger</propdesc>
  </property>
  <property>
    <propvalue>green</propvalue>
    <propdesc>depicts permission</propdesc>
  </property>
</properties>

3.2.4.6 `<prophhead>`
The `<prophhead>` element supports headings for the `<properties>` element.

Specialization hierarchy
The `<prophhead>` element is specialized from the `<sthead>` element in the topic module.

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
See the example in properties (81).

3.2.4.7 `<proptypehd>`
The `<proptypehd>` element supports headings for the type column of a properties table.

Specialization hierarchy
The `<proptypehd>` element is specialized from the `<stentry>` element in the topic module.

Attributes
STUB CONTENT
Example
See the example in properties (81).

3.2.4.8 <propvaluehd>
The <propvaluehd> element supports headings for the value column of a properties table.

Specialization hierarchy
The <propvaluehd> element is specialized from the <stentry> element in the topic module.

Attributes
STUB CONTENT

Example
See the example in properties (81).

3.2.4.9 <propdescrd>
The <propdescrd> element supports headings for the description column of a properties table.

Specialization hierarchy
The <propdescrd> element is specialized from the <stentry> element in the topic module.

Attributes
STUB CONTENT

Example
See the example in properties (81).

3.2.4.10 <property>
The <property> element represents a single property of the current topic's subject. For example, if the current reference topic describes a programming class, the property might show that the class is protected rather than public.

Usage information
The <property> element generally appears together with a series of other properties; it contains three optional elements to provide a type, value, or description of the property.

Specialization hierarchy
The <property> element is specialized from the <strow> element in the topic module.

Attributes
The following attributes are available on this element: Universal attribute group (159).
Example
See properties (81).

3.2.4.11 <proptype>
The <proptype> element describes the type of the property.

Specialization hierarchy
The <proptype> element is specialized from the <stentry> element in the topic module.

Attributes
STUB CONTENT

Example
See properties (81).

3.2.4.12 <propvalue>
The <propvalue> element indicates one or more values for the current property type.

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

Specialization hierarchy
The <propvalue> element is specialized from the <stentry> element in the topic module.

Attributes
STUB CONTENT

Example
See properties (81).

3.2.4.13 <propdesc>
The <propdesc> element is used to provide a short description of the property type and its listed values.

Specialization hierarchy
The <propdesc> element is specialized from the <stentry> element in the topic module.

Attributes
STUB CONTENT

Example
See properties (81).
3.2.5 Task elements
Task topics answer "How do I?" questions, and have a well-defined structure that describes how to complete a procedure to accomplish a specific goal. Use the task topic to describe the steps of a particular task, or to provide an overview of a higher-level task. The task topic includes sections for describing the context, prerequisites, actual steps, expected results, example, and expected next steps for a task.

3.2.5.1 <chdesc>
The <chdesc> element provides the content of the "Description" cell in a choice table row. It explains why the user would choose that option and might explain the result of the choice when it is not immediately obvious.

Specialization hierarchy
The <chdesc> element is specialized from <stentry>. It is defined in the task module.

Attributes

Example
See choicetable (86).

3.2.5.2 <chdeschd>
The <chdeschd> element provides a label for the second ("Description") column in a choice table.

Specialization hierarchy
The <chdeschd> element is specialized from <stentry>. It is defined in the task module.

Attributes

Example
See choicetable (86).

3.2.5.3 <chhead>
The <chhead> element contains optional header row content for a choice table.

Rendering expectations
Labels provided by the <chhead> element override any default headings for the <choicetable> that are provided by stylesheets.

Specialization hierarchy
The <chhead> element is specialized from <sthead>. It is defined in the task module.

Attributes
Example
See choicetable (86).

3.2.5.4 <choice>
A <choice> element describes a way that a user can complete the current step.

Specialization hierarchy
The <choice> element is specialized from <li>. It is defined in the task module.

Attributes

Example
See choices (86)

3.2.5.5 <choices>
The <choices> element contains a list of choices. Each choice represents a way that a user can complete the current step.

Specialization hierarchy
The <choices> element is specialized from <ul>. It is defined in the task module.

Attributes

Example
The following code sample shows how a user can be presented with different ways to select a server:

```xml
<step>
  <cmd>Choose a server:</cmd>
  <choices>
    <choice>If you have a remote server you want to test on, type the IP address or hostname of the server.</choice>
    <choice>If you want to do local testing, type localhost.</choice>
  </choices>
</step>
```

3.2.5.6 <choicetable>
The <choicetable> element contains information about a set of options that are presented to users in a step.

Specialization hierarchy
The <choicetable> element is specialized from <simpletable>. It is defined in the task module.

Attributes
The following attributes are available on this element: STUB CONTENT, STUB CONTENT, STUB CONTENT (with a narrowed definition of @keycol, given below), #unique_24/unique_24_Connect_42_outputclass, and #unique_24/unique_24_Connect_42_spectitle.
@keycol
Defines the column that contains headings for each row. The default value for @keycol on <choicetable> is "1". Otherwise, the attribute is the same as defined in STUB CONTENT.

Example
The following code sample contains a choice table that contains a header row:

```xml
<step>
  <cmd>Create a new filter:</cmd>
  <choicetable>
    <chhead>
      <choptionhd>Option</choptionhd>
      <chdeschd>Description</chdeschd>
    </chhead>
    <chrow>
      <choption>Command-line interface</choption>
      <chdesc>Type arg -f filter</chdesc>
    </chrow>
    <chrow>
      <choption>Product GUI</choption>
      <chdesc>Click New Filter</chdesc>
    </chrow>
  </choicetable>
</step>
```

3.2.5.7 <choption>
The <choption> element contains the content of the "Option" cell in a choice table row.

Specialization hierarchy
The <choption> element is specialized from <stentry>. It is defined in the task module.

Attributes
STUB CONTENT

Example
See choicetable (86).

3.2.5.8 <choptionhd>
<choptionhd> element provides a label for the first ("Options") column in a choice table.

Specialization hierarchy
The <choptionhd> element is specialized from <stentry>. It is defined in the task modules.

Attributes

Example
See choicetable (86).
3.2.5.9 <chrow>
The <chrow> element represents a row in a choice table. It contains a pair of elements: <choption> and <chdesc>.

**Specialization hierarchy**
The <strow> element is specialized from <strow>. It is defined in the task module.

**Attributes**
The following attributes are available on this element: Universal attribute group (159).

**Example**
See choicetable (86).

3.2.5.10 <cmd>
The <cmd> element contains a command. A command provides instruction on how to complete a step.

**Specialization hierarchy**
The <cmd> element is specialized from <ph>. It is defined in the task module.

**Attributes**
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

**Example**
In the following code sample, the <cmd> element provides clear, active-voice instruction for how to complete a step.

```xml
<step>
  <cmd>Specify the configuration parameters.</cmd>
</step>
```

3.2.5.11 <context>
The <context> element contains contextual information. Contextual information is background information that helps users understand the purpose of the task and what they will gain by completing it.

**Specialization hierarchy**
The <context> element is specialized from <section>. It is defined in the task module.

**Attributes**
The following attributes are available on this element: Universal attribute group (159).
Example
An author uses the following markup to provide users with more contextual information than is appropriate for a short description. Style sheets might generate a label, for example, "About this procedure", to indicate clearly that the information provided is background information.

```xml
<task id="Generating_stub_files" xml:lang="en-us">
  <title>Generating Stub Files</title>
  <shortdesc>You can use IBM Task Modeler to generate stub files. Stub files are DITA files that contain only a title.</shortdesc>
  <taskbody>
    <context>As you perform this procedure, you can select the conventions that you want to use for file names.</context>
    ...
  </taskbody>
</task>
```

3.2.5.12 <info>
The <info> element contains additional information about the step.

Specialization hierarchy
The <info> element is specialized from <itemgroup>. It is defined in the task module.

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
In the following code sample, the <info> element provides additional information about the ways that the step can be performed.

```xml
<step>
  <cmd>Specify the configuration parameters.</cmd>
  <info>The configuration parameters can be specified from either the command line or the product GUI.</info>
</step>
```

3.2.5.13 <postreq>
The <postreq> element contains post-requisites. Post-requisites are steps or tasks that the users need to perform after completing the current task.

Specialization hierarchy
The <postreq> element is specialized from <section>. It is defined in the task module.

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
The following code sample shows how a user might be directed to notify a test proctor after completing a test.

```xml
<postreq>Notify the proctor upon completing this self-test.</postreq>
```
3.2.5.14 <prereq>
The <prereq> contains prerequisites. Prerequisites are things that users need to know or preliminary tasks that users need to perform before starting the current task.

Rendering expectations
Processors MAY render prerequisite links from the related-links section together with the content of the <prereq> element.

Specialization hierarchy
The <prereq> element is specialized from <section>. It is defined in the task module.

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
The following code sample is from a topic that explains how to create an SQLJ file. A prerequisite is to log into the SQLJ server. Style sheets might generate a label, for example, “Before you begin”, to indicate clearly that the prerequisite task needs to be performed before embarking on the procedure.

```xml
<task id="sqlj">
<title>Creating an SQLJ file</title>
<taskbody>
  <prereq>Before creating a new SQLJ file, you must log in to the SQLJ server.</prereq>
...
</taskbody>
</task>
```

3.2.5.15 <result>
The <result> element contains the end result. A result describes the expected outcome for the task as a whole.

Specialization hierarchy
The <result> element is specialized from <section>. It is defined in the task module.

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
In the following code sample, the author clearly communicates the expected result of successfully completing the task:

```xml
<task id="sqlj">
<title>Creating an SQLJ file</title>
<taskbody>
  <result>The "File Created" window is displayed, and the SQLJ file is successfully created.</result>
</taskbody>
</task>
```
3.2.5.16 <step>
The <step> element contains a step. A step represents an action that a user can take to complete a task. It also can contain additional information about the step, such as an example, result, or trouble shooting guidance.

Specialization hierarchy
The <step> element is specialized from <li>. It is defined in the task module.

Attributes
The following attributes are available on this element: Universal attribute group (159) (with a narrowed definition of @importance, given below) and outputclass (159).

importance
Specifies whether the step is optional or required. Output processors might highlight steps that are optional or required. Allowed values are optional, required, or -dita-use-conref-target (177).

Example
The following code sample shows almost all the elements that <step> can contain.

```xml
<step>
  <cmd>Specify the configuration parameters.</cmd>
  <info>The configuration parameters can be specified from either the command line or the product GUI.</info>
  <choices>
    <choice>From a command prompt, type config -l parameter</choice>
    <choice>Click New Configuration Parameters.</choice>
  </choices>
  <stepresult>You receive a 'Configuration successful message".</stepresult>
  <steptroubleshooting>If you do not receive a 'Configuration successful message," retry the configuration operation.</steptroubleshooting>
</step>
```

3.2.5.17 <stepresult>
The <stepresult> element provides information about the expected outcome of a step.

Specialization hierarchy
The <stepresult> element is specialized from <itemgroup>. It is defined in the task module.

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
In the following example, the content of the <stepresult> element enables the user to ascertain whether they have completed the step correctly.

```xml
<step>
  <cmd>Specify the configuration parameters.</cmd>
  <info>The configuration parameters can be specified from either the command line or the product GUI.</info>
</step>
```
3.2.5.18 <steps>
The <steps> element contains a set of steps. Steps are a series of actions that are conducted in a specific order and manner.

Specialization hierarchy
The <steps> element is specialized from <ol>. It is defined in the task module.

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
The following code sample shows a simple task topic with two steps:

```xml
<task id="sqlj">
  <title>Creating an SQLJ file</title>
  <taskbody>
    <context>Once you have set up SQLJ, you need to create a new SQLJ file.</context>
    <steps>
      <step>
        <cmd>In a text editor, create a new file.</cmd>
      </step>
      <step>
        <cmd>Enter the first query statement.</cmd>
      </step>
    </steps>
  </taskbody>
</task>
```

3.2.5.19 <steps-informal>
The <steps-informal> element contains a set of informal steps. Informal steps do not follow a strict content model; a paragraph might describe more than one step, or a paragraph might combine procedural information along with other information.

Specialization hierarchy
The <steps-informal> element is specialized from <section>. It is defined in the task module.

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
An author uses the following markup to provide informal information about how to grow a flower from seed:

```xml
<task>
  <title>Growing a flower from seed</title>
  <taskbody>
```
Put the soil in the container any old way. It doesn't really matter how you do it as long as it is at least 12 cm deep. Once the soil is in place, plant the seeds, water appropriately and wait.

3.2.5.20 <steps-unordered>
The <steps-unordered> element contains a set of unordered steps. Unordered steps are steps in which the order of the steps to be performed might vary from one situation to another.

**Specialization hierarchy**
The <steps-unordered> element is specialized from <ul>. It is defined in the task module.

**Attributes**
The following attributes are available on this element: Universal attribute group (159).

**Example**
A author uses the following markup to indicate tasks that need to be performed before leaving on a vacation. While each step involves a single item, the steps can be performed in any order.

```xml
<task id="task">
  <title>Preparing for a trip</title>
  <taskbody>
    <steps-unordered>
      <step>
        <cmd>Arrange for a pet sitter</cmd>
      </step>
      <step>
        <cmd>Do laundry</cmd>
      </step>
      <step>
        <cmd>Buy a plane ticket</cmd>
      </step>
    </steps-unordered>
  </taskbody>
</task>
```

3.2.5.21 <stepsection>
The <stepsection> element contains expository text that might be rendered before a step.

**Rendering expectations**
Processors which render the content of <stepsection> elements among the <step> elements MUST NOT number the <stepsection> elements.

**Specialization hierarchy**
The <stepsection> element is specialized from <li>. It is defined in the task module.

**Attributes**
The following attributes are available on this element: Universal attribute group (159).
Example
The following code sample illustrates how a `<stepsection>` element might be used:

```xml
<steps>
    <step>
        <cmd>Get out a bowl.</cmd>
    </step>
    <stepsection>The next two steps are very important!</stepsection>
    <step>
        <cmd>Put on safety gloves.</cmd>
    </step>
    <step>
        <cmd>Put on goggles.</cmd>
    </step>
    <step>
        <cmd>Pour milk and cereal into the bowl.</cmd>
    </step>
</steps>
```

The code sample above typically would be rendered with "Get out a bowl" as step number one, "Put on safety gloves" as step number two, and "The next two steps are very important!" as an unnumbered item in between the first two items.

3.2.5.22 `<steptroubleshooting>`
The `<steptroubleshooting>` element contains information that is designed to help remedy the situation when a step does not complete as expected.

Usage information
Tip  Do not use `<note type="trouble">` inside of the `<steptroubleshooting>` element, because its meaning there would be ambiguous.

Specialization hierarchy
The `<steptroubleshooting>` element is specialized from `<itemgroup>`. It is defined in the task module.

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
The following code sample shows using the `<steptroubleshooting>` element with several actions:

```xml
<step>
    <cmd>Log in to the system</cmd>
    <stepresult>
        <p>The `<wintitle>Welcome</wintitle>` screen appears.</p>
    </stepresult>
    <steptroubleshooting>
        <p>If the `<wintitle>Welcome</wintitle>` screen does not appear, try one or more of the following:</p>
        <ul>
            <li>Verify that the user name was entered correctly</li>
            <li>Verify that the password was entered correctly</li>
            <li>Confirm that the maintenance contract is still active</li>
        </ul>
    </steptroubleshooting>
</step>
```
3.2.5.23 `<stepxmp>`

The `<stepxmp>` element contains an example that illustrates how a step is completed. This example can be a few words, an image, or a paragraph.

**Specialization hierarchy**

The `<stepxmp>` element is specialized from `<itemgroup>`. It is defined in the task module.

**Attributes**

The following attributes are available on this element: Universal attribute group (159).

**Example**

In the following code sample, the `<stepxmp>` element contains an example of what a user might type as a name for a widget:

```xml
<step>
  <cmd>Type a name for the widget.</cmd>
  <stepxmp>For example, <userinput>mywidget</userinput></stepxmp>
</step>
```

3.2.5.24 `<task>`

The `<task>` element is the top-level element for a task topic. Tasks topics provide step-by-step instructions that enable a user to perform a task; they tell the user precisely what to do and the order in which to do it.

**Usage information**

The OASIS DITA Technical Committee distributes two document-type shells for tasks: general task and strict task.

**(Strict) task**

Maintains the order and cardinality of the original `<taskbody>` content model. The strict task is implemented with a constraint module.

**General task**

Relaxes the content model of the original task topic. It allows `<section>` and `<steps-informal>` inside of the task body; it also allows multiple instances and varying order for the elements that make up the task body.

**Specialization hierarchy**

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

**Attributes**

STUB CONTENT
Example

The following code sample illustrates the main building blocks of a strict task topic:

```xml
<task id="Generating-stub-files" xml:lang="en-us">
  <shortdesc>You can use IBM Task Modeler to generate stub files. Stub files are DITA files that contain only a title.</shortdesc>
  <taskbody>
    <prereq>You must have created a DITA map in IBM Task Modeler.</prereq>
    <context>As you perform this procedure, you can select the conventions that you want to use for file names.</context>
    <steps>...
    </steps>
    <result>In the File Manager view, you can see the file names and paths of the DITA topics.</result>
    <tasktroubleshooting>If you cannot see the file name and paths of the DITA topics, refresh the view.</tasktroubleshooting>
    <example>...
    </example>
    <postreq>You now can create a relationship table to define links between the topics in your DITA map.</postreq>
  </taskbody>
</task>
```

In a strict task topic, while the child elements of `<taskbody>` are all optional, they can only occur once and must appear in a specific order.

3.2.5.25 `<taskbody>`

The `<taskbody>` element contains the body of a task topic. The task body can include prerequisites, contextual information, steps, results, an example, troubleshooting information, and post-requisites.

Usage information

The content model of the `<taskbody>` element depends on the type of task topic used.

Specialization hierarchy

The `<taskbody>` element is specialized from `<body>`. It is defined in the task module.

Attributes

STUB CONTENT

Example

See task (95).

3.2.5.26 `<tasktroubleshooting>`

The `<tasktroubleshooting>` element contains troubleshooting information that is designed to help users remedy the situation when a task does not complete as expected.

Usage information

Tip Do not use `<note type="trouble">` inside of the `<tasktroubleshooting>` element, because its meaning there would be ambiguous.
Specialization hierarchy

The `<tasktroubleshooting>` element is specialized from `<section>`. It is defined in the task module.

Attributes

The following attributes are available on this element: Universal attribute group (159).

Example

In the following code sample, the `<tasktroubleshooting>` element contains brief information that explains the steps that the user can take when the results of a task are not as expected. For a complex remedy, the author could provide a link to another task topic.

```
<task>
  <title>Adding new user categories</title>
  <taskbody>
    <steps>...
    </steps>
  </taskbody>
  <result>
    <p>The User Type menu displays the new types you added.</p>
  </result>
  <tasktroubleshooting>
    <p>If the User Type menu does not display the additions, try one or more of the following:
    </ul>
    <li>Refresh the page</li>
    <li>Verify that Add Types window is not still open; if so, go to it and press <uicontrol>OK</uicontrol>.</li>
  </ul>
  </p>
</tasktroubleshooting>
```

3.2.5.27 <tutorialinfo>

The `<tutorialinfo>` element contains additional information that is useful when the task topic is part of a tutorial.

Specialization hierarchy

The `<tutorialinfo>` element is specialized from `<itemgroup>`. It is defined in the task module.

Attributes

The following attributes are available on this element: Universal attribute group (159).

Example

The following code sample shows how the `<tutorialinfo>` element might be used in a task topic that is part of a tutorial.

```
<steps>
  <step>
    <cmd>Do this</cmd>
    <tutorialinfo>In your editor, open the first element and click on the dialog.</tutorialinfo>
  </step>
  <step>
    <cmd>Do that</cmd>
  </step>
```
3.2.6 Troubleshooting elements
Troubleshooting topics document corrective action such as troubleshooting or alarm clearing.

3.2.6.1 <troubleshooting>
The <troubleshooting> element is the top-level element for a troubleshooting topic.

Usage information
Troubleshooting topics begin with a description of a condition that the reader might want to correct, followed by one or more cause-remedy pairs. Each cause-remedy pair is a potential solution to the trouble described in the condition. Troubleshooting topics represent the kind of information that users typically consult to fix a problem.

Specialization hierarchy
- topic/topic troubleshooting/troubleshooting

Attributes

Example

```xml
<troubleshooting id="resolve" xml:lang="en-us">
  <title>E247 - Memory fault has occurred</title>
  <shortdesc>The system has detected a problem in memory.</shortdesc>
  <troublebody>
    <condition>
      <p>The fault indicator flashes on the front panel, and the error log contains the following message:
      <msgph>E247 - Memory fault has occurred</msgph></p>
    </condition>
    <troubleSolution>
      <cause>
      <p>Transient memory fault.</p>
      </cause>
      <remedy>
        <responsibleParty>System administrator</responsibleParty>
        <steps>
          <step>
            <cmd>Reset the alarm</cmd>
          </step>
          <step>
            <cmd>Monitor the system periodically to see whether the alarm recurs</cmd>
          </step>
        </steps>
      </remedy>
    </troubleSolution>
    <troubleSolution>
      <cause>
      <p>Recurring memory fault indicates possible problem with the system memory board. Reseating the board may fix the problem.</p>
      </cause>
      <remedy>
        <steps>
          ...
        </steps>
      </remedy>
    </troubleSolution>
  </troublebody>
</troubleshooting>
```
3.2.6.2 <troublebody>
The <troublebody> element is a container for the main content of the troubleshooting topic. Troubleshooting topics limit the body structure to a single optional condition followed by one or more solutions.

Inheritance
- topic/body troubleshooting/troublebody

Example
See troubleshooting (98).

Attributes
3.2.6.3 <condition>
The <condition> element describes a state that the troubleshooting topic is intended to remedy. This information helps the user decide whether a troubleshooting topic might contain an applicable remedy for a problem.

Usage information
This section should add to or clarify information that is in the tile or short description of the troubleshooting topic.

Specialization hierarchy
- topic/section troubleshooting/condition
Example
See troubleshooting (98).

Attributes

3.2.6.4 `<troubleSolution>`
The `<troubleSolution>` element is a container element for cause and remedy information.

Usage information
The cause might be omitted if it is implicit or if the remedy is not associated with a cause. The remedy might be omitted if there is no known remedy for the cause.

Specialization hierarchy
- topic/section troubleshooting/troubleSolution

Attributes

Example
See troubleshooting (98).

3.2.6.5 `<cause>`

Usage information
This information should be brief; if it is not possible to make the information brief, this might be a sign that a full troubleshooting topic is needed.

Specialization hierarchy
- topic/section troubleshooting/cause

Example
See troubleshooting (98).

Attributes

3.2.6.6 `<remedy>`
The `<remedy>` element contains steps that are a potential solution for the problem described in the `<condition>` element, the `<title>` element, or the `<shortdesc>` element.

Usage information
Typically, a `<remedy>` is preceded by a `<cause>`.

Specialization hierarchy
- topic/section troubleshooting/remedy
3.2.6.7 <responsibleParty>
The <responsibleParty> element identifies the individual or team whose task it is to perform a remedy procedure.

Usage information

Specialization hierarchy
- topic/p troubleshooting/responsibleParty

Example
See troubleshooting (98).

Attributes

3.3 Domain specializations
Domains in this section include those generally associated with technical content, such as the programming and software domains.

3.3.1 Equation domain
The elements in the equation domain enable authors to clearly distinguish equations from other type of content. These markup distinctions can enable formatting distinctions, numbering of equations, and more. This domain can be used independently of the MathML domain.

The equation domain elements do not attempt to satisfy all possible requirements for representing equations semantically. More sophisticated users of equations might need to further specialize from these elements or provide an alternative vocabulary for semantic equations.

3.3.1.1 <equation-block>
Use the <equation-block> element to represent an equation that is presented as a separate block within a text flow.

Usage information

When an <equation-block> element has multiple direct child elements, each child represents an alternative form of the equation.

Formatting expectations
Block equations can be numbered.
Processing expectations

Processors are free to choose the form or forms that they use in deliverables. For example, if there is both an image and MathML markup, an HTML-generating processor could output both the image reference and the MathML with appropriate HTML @class or @id values to enable dynamic showing or hiding of one form or the other based on browser capability.

Specialization hierarchy

+ topic/divequation-d/equation-block

Attributes

The following attributes are available on this element: Universal attribute group (159).

Example

In the following example, a block equation uses MathML as its content:

```
<p>A block equation using MathML:</p>
<equation-block>
  <mathml>
    <m:math>
      <m:semantics>
        <m:mrow>
          <m:msqrt>
            <m:mrow>
              <m:msup>
                <m:mi>a</m:mi>
                <m:mn>2</m:mn>
              </m:msup>
              <m:mo>+</m:mo>
              <m:msup>
                <m:mi>b</m:mi>
                <m:mn>2</m:mn>
              </m:msup>
            </m:mrow>
          </m:msqrt>
        </m:mrow>
      </m:semantics>
    </m:math>
  </mathml>
</equation-block>
```

In the following example, a block equation uses an image as its content:

```
<p>A block equation using an image:</p>
<equation-block>
  <image keyref="equation-image-01">
    <alt>a squared plus b squared.</alt>
  </image>
</equation-block>
```

In the following example, the block equation contains two alternative forms of the same equation:

```
<equation-block>
  <image keyref="equation-image-01">
    <alt>a squared plus b squared.</alt>
  </image>
  <mathml>
    <m:math>
      <m:semantics>
        <m:mrow>
          <m:msqrt>
            <m:mrow>
              <m:msup>
                <m:mi>a</m:mi>
                <m:mn>2</m:mn>
              </m:msup>
              <m:mo>+</m:mo>
              <m:msup>
                <m:mi>b</m:mi>
                <m:mn>2</m:mn>
              </m:msup>
            </m:mrow>
          </m:msqrt>
        </m:mrow>
      </m:semantics>
    </m:math>
  </mathml>
</equation-block>
```
3.3.1.2 <equation-figure>

Use the <equation-figure> element to represent an equation that functions as a form of figure or display.

Usage information

Since the <equation-figure> element is specialized from <fig>, display equations can have titles, descriptions, figure groups, and all other figure components. The direct children of <equation-figure> can be the equation content itself (for example, <mathml> or an image reference), or it can be one or more <equation-block> elements, along with other elements allowed within <fig>, such as paragraphs.

When an <equation-figure> element has multiple direct child <mathml>, <image>, or <pre> elements, each child represents an alternative form of the equation.

When the intent is to have equations combined with other commentary within an <equation-figure>, the recommended best practice is to use child <equation-block> elements to contain the equations and clearly distinguish them from the commentary.

Formatting expectations

Display equations are intended to be numbered when numbering is desired.

Processing expectations

Processors are free to choose the form or forms that they use in deliverables. For example, if there is both an image and MathML markup, an HTML-generating processor could output both the image reference and the MathML with appropriate HTML @class or @id values to enable dynamic showing or hiding of one form or the other based on browser capability. All other direct-child elements of <equation-figure> are treated normally.

Specialization hierarchy

+ topic/fig equation-d/equation-figure

Attributes

The following attributes are available on this element: Universal attribute group (159), Display attribute group (166), and spectitle (169).
Example

In the following code sample, the <equation-figure> element contains a title and MathML:

```xml
<equation-figure>
  <title>Display equation with a MathML container</title>
  <mathml>
    <m:math display='block'>
      <m:semantics>
        <m:mrow>
          <m:mfrac>
            <m:mi>n</m:mi><m:mo>!</m:mo>
            <m:mrow><m:mo>(</m:mo>
              <m:mrow>
                <m:mi>n</m:mi><m:mo>&#x2212;</m:mo><m:mi>r</m:mi>
              </m:mrow><m:mo>)</m:mo></m:mrow><m:mo>!</m:mo>
          </m:mrow>
        </m:semantics>
      </m:math>
    </mathml>
</equation-figure>
```

In the following example, the <equation-figure> element contains a title and an <equation-block> element that contains MathML and commentary:

```xml
<equation-figure>
  <title>Display equation with a MathML container</title>
  <equation-block>
    <mathml>
      <m:math display='block'>
        <m:semantics>
          <m:mrow>
            <m:mfrac>
              <m:mi>n</m:mi><m:mo>!</m:mo>
              <m:mrow><m:mo>(</m:mo>
                <m:mrow>
                  <m:mi>n</m:mi><m:mo>&#x2212;</m:mo><m:mi>r</m:mi>
                </m:mrow><m:mo>)</m:mo></m:mrow><m:mo>!</m:mo>
            </m:mrow>
          </m:semantics>
        </m:math>
      </mathml>
    </equation-block>
    <p>Where <equation-inline><mathml><m:math><m:mi>r</m:mi></m:math></mathml></equation-inline> is greater than 1.</p>
</equation-figure>
```
3.3.1.3 <equation-inline>
Use the <equation-inline> element to represent an equation that is presented inline within a paragraph or similar context.

Usage information
Inline equations are not intended to be numbered.
When an <equation-inline> element has multiple direct child elements, each child represents an alternative form of the equation.

Processing expectations
Processors are free to choose the form or forms that they use in deliverables. For example, if there is both an image and MathML markup, an HTML-generating processor could output both the image reference and the MathML with appropriate HTML @class or @id values to enable dynamic showing or hiding of one form or the other based on browser capability.

Specialization hierarchy
+ topic/ph equation-d/equation-inline

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
The following example contains a paragraph that contains an <equation-inline> element with MathML markup:

```
<p>MathML inline: <equation-inline>
  <mathml>
    <m:math display='inline'>
      <m:semantics>
        <m:mrow>
          <m:msqrt>
            <m:mrow>
              <m:msup>
                <m:mi>a</m:mi>
                <m:mn>2</m:mn>
              </m:msup>
              <m:mo>+</m:mo>
              <m:msup>
                <m:mi>b</m:mi>
                <m:mn>2</m:mn>
              </m:msup>
            </m:mrow>
          </m:msqrt>
        </m:mrow>
      </m:semantics>
    </m:math>
  </mathml>
</equation-inline>
```

The following example contains an <equation-inline> element that uses an image as its content:

```
<p>An inline equation that contains an image:
  <equation-inline>
    <image keyref="equation-image-01">
      <alt>a squared plus b squared.</alt>
  </image>
</equation-inline>
```
3.3.1.4 <equation-number>
Use the <equation-number> element to indicate that a block equation is numbered and, optionally, to specify the number to use for the block equation.

Usage information
In normal usage a block equation will have at most one number. However, <equation-number> can occur multiple times within <equation-block>, for example, to allow numbers with different (and exclusive) conditional properties.

Formatting expectations
When the <equation-number> element has empty or whitespace-only content, the equation number SHOULD be generated. When the <equation-number> element has non-whitespace-only content the content SHOULD be used as the equation number. For explicit numbers the content of the element SHOULD be the number value without any surrounding punctuation, for example, "3.2a" rather than "(3.2a)". Processors SHOULD add punctuation or decoration to the number as appropriate.

The details of equation numbering and number presentation are processor-specific. However, common practice is to present the equation number to the right of the equation, centered vertically within the vertical extent of the block equation.

Specialization hierarchy
+ topic/ph equation-d/equation-number

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
An equation with a generated number:

```xml
<p>A block equation using an image:</p>
<equation-block id="eq-001">
  <equation-number/>
  <image keyref="equation-image-01">
    <alt>a squared plus b squared.</alt>
  </image>
</equation-block>
```

An equation with an explicit number:

```xml
<p>A block equation using an image:</p>
<equation-block id="eq-3.2a">
  <equation-number>3.2a</equation-number>
  <image keyref="equation-image-01">
    <alt>a squared plus b squared.</alt>
  </image>
</equation-block>
```
3.3.2 Markup domain
The markup domain elements are used for the mention of named constructs in markup languages, such as XML.

3.3.2.1 <markupname>
The <markupname> element identifies named markup tokens, for example, elements or attributes in XML and SGML, named groups in XSD schemas, and named patterns in RELAX NG schemas.

Usage information
The <markupname> element serves as the specialization basis for the elements in the XML mention domain. When the XML mention domain is present, use its more specific elements instead of the <markupname> element when appropriate.

Specialization hierarchy
The <markupname> element is specialized from <keyword>. It is defined in the markup domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
In the following code sample, the <markupname> element is used to tag an attribute group.

The <markupname>p.attributes</markupname> attribute group defines the allowed attributes for the <xmlelement>p</xmlelement> element.

3.3.3 MathML domain
The MathML domain elements enable direct use of MathML markup within DITA documents, as well as use-by-reference of MathML markup that is stored separate, non-DITA documents. MathML is a W3C standard.

For MathML markup that is stored directly in DITA documents that are validated using DTDs, the MathML elements must use a namespace prefix in order to avoid conflict with the DITA-defined elements of the same name. Documents validated using XSD or RELAX NG can default the MathML namespace on the MathML <math> element. MathML elements that are referenced using the <mathmlref> element do not need to have a namespace prefix, because they are parsed separately from the DITA documents that refer to them. By default, the MathML domain is configured to use the namespace prefix "m" for the MathML elements.

Related information
Mathematical Markup Language (MathML), Version 3.0
3.3.3.1 <mathml>
Use the <mathml> element to contain content that contributes to a semantic equation.

Usage information
Mathml element content includes MathML elements, references to MathML elements held in separate, non-DITA documents, <data>, or <data-about> elements.

The <mathml> element is not intended to represent a semantic equation, only content that contributes to a semantic equation. Use the equation domain elements or their equivalent to represent equations semantically, for example, to enable numbering of equations.

The MathML markup must have a root element of <math> within the MathML namespace: "http://www.w3.org/1998/Math/MathML".

Specialization hierarchy
+ topic/foreign mathml-d/mathml

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
In the following code sample, a <mathml> element contains MathML content:

```xml
<equation-block>
  <mathml>
    <m:math xmlns:m="http://www.w3.org/1998/Math/MathML"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      <m:semantics>
        <m:mrow>
          <m:msqrt>
            <m:mrow>
              <m:msup>
                <m:mi>a</m:mi>
                <m:mn>2</m:mn>
              </m:msup>
              <m:mo>+</m:mo>
              <m:msup>
                <m:mi>b</m:mi>
                <m:mn>2</m:mn>
              </m:msup>
            </m:mrow>
          </m:msqrt>
        </m:mrow>
      </m:semantics>
    </m:math>
  </mathml>
</equation-block>
```

3.3.3.2 <mathmlref>
Use the <mathmlref> element to reference a non-DITA XML document that contains MathML markup.

Usage information
The <mathmlref> element enables you to use the markup by reference.

The reference must be to a MathML <math> element. The reference can be one of the following:
A URI that addresses an XML document; the XML document has a MathML `<math>` element as the root element
A URI that addresses an XML document and contains a fragment identifier that is the XML ID of a `<math>` element within the document

The reference can be direct, using the `@href` attribute, or indirect, using the `@keyref` attribute. For indirect referencing, specify only the key name. Specify the ID of the `<math>` element as part of the value for the `@href` attribute on the key definition.

For example, to refer to the `<math>` element with the `@id` of "equation-02" within a larger document using a key reference, you would define the key in the following way:

```
<keydef
  keys="mathml-equation-02"
  href="math/mathml-equations.xml#equation-02"
  format="mathml"
/>
```

You would refer to this key using just the key name:

```
<mathml>
  <mathmlref keyref="mathml-equation-02"/>
</mathml>
```

Processing expectations

Processors **SHOULD** process the MathML as though the `<m:math>` element had occurred directly in the content of the containing `<mathml>` element.

Specialization hierarchy

+ topic/xref mathml-d/mathmlref

Attributes

The following attributes are available on this element: **STUB CONTENT**, `#unique_24/unique_24_Connect_42_outputclass`, and `@keyref`. This element also uses `@href`, `@scope`, and a narrowed definition of `@format` (given below) from **STUB CONTENT**.

@format

Specifies the format (data type) of the referenced resource. For MathML the format should be "mml", which is the default value for `@format` on this element.

Example

Here is a reference to a `<mathml>` element that is the root element of its containing document:

```
<equation-block>
  <mathml>
    <mathmlref href="../mathml-source/mathml-root-mathml.mml"/>
  </mathml>
</equation-block>
```
The `mathml-root-mathml.mml` file contains the following content. Note that the `<math>` element sets the MathML namespace as the default namespace, so there are no namespace prefixes on the MathML markup:

```xml
<?xml version="1.0" encoding="UTF-8"/>
<math xmlns="http://www.w3.org/1998/Math/MathML"
xmlns:xlink="http://www.w3.org/1999/xlink">
<mstyle displaystyle="false" scriptlevel="0">
<mrow>
<mfrac>
<mrow>
<mi mathcolor="gray">sin</mi>
<mo rspace="verythinmathspace"></mo>
<mi>θ</mi>
</mrow>
<mi>π</mi>
</mfrac>
</mrow>
</mstyle>
</math>
```

Here is a reference to a specific `<math>` element in a containing XML file:

```xml
<equation-block>
<mathml>
<mathmlref href="../mathml-source/mathml-equation-library.xml#mathfrag-02"/>
</mathml>
</equation-block>
```

The `mathml-equation-library.xml` file contains the following content:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<root>
<part>
<math id="timeinday"
 xmlns="http://www.w3.org/1998/Math/MathML">
<mi>x</mi>
</math>
<math id="mathfrag-02"
 xmlns="http://www.w3.org/1998/Math/MathML">
<mrow>
<mi>y</mi><mo>=</mo><mn>5</mn><mi>x</mi><mo>+</mo><mn>2</mn>
</mrow>
</math>
</part>
<part>
<m:math id="mathfrag-03"
 display="inline"
 overflow="scroll"
 xmlns:m="http://www.w3.org/1998/Math/MathML">
<m:mi>y</m:mi><m:mo>=</m:mo><m:mi>m</m:mi><m:mo>x</m:mo><m:mo>+</m:mo><m:mi>b</m:mi>
</m:math>
</part>
</root>
```
3.3.4 Programming domain
The programming domain elements are used to define the syntax for programming languages. They also can be used to provide examples.

3.3.4.1 <apiname>
The <apiname> element identifies the name of an application programming interface (API), such as a Java class name or method name.

Usage information
This element is part of the programming domain.

Specialization hierarchy
The <apiname> is specialized from <keyword>. It is defined in the programming domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
In the following code sample, the <apiname> element identifies the document.write method.

```
<p>Use the <apiname>document.write</apiname> method to create text output in the dynamically constructed view.</p>
```

3.3.4.2 <codeblock>
The <codeblock> element identifies lines of program code.

Usage information
This element is part of the programming domain.

Rendering expectations
Processors SHOULD preserve line the breaks and spaces that are present in the content of a <codeblock> element.

Specialization hierarchy
The <codeblock> is specialized from <pre>. It is defined in the programming domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159), Display attribute group (166), xml:space (168), and spectitle (169).
Example
In the following code sample, the `<codeblock>` element contains an excerpt from the code for a program:

```xml
<codeblock>
/* a long sample program */
Do forever
  Say "Hello, World"
End
</codeblock>
```

### 3.3.4.3 `<codeph>`
The `<codeph>` element identifies a code snippet.

#### Usage information
This element is part of the programming domain.

#### Specialization hierarchy
The `<codeph>` is specialized from `<ph>`. It is defined in the programming domain module.

#### Attributes
The following attributes are available on this element: [Universal attribute group](#) (159) and [@keyref](#) (180).

Example
In the following code sample, the `<codeph>` element identifies a code snippet. The code snippet will be rendered in-line in the paragraph.

```
<p>The second line of the sample program code, `<codeph>Do forever</codeph>`, represents the start of a loop construct.</p>
```

### 3.3.4.4 `<coderef>`
The `<coderef>` element references an external file that contains literal code.

#### Usage information
This element is part of the programming domain.

#### Rendering expectations
When evaluated, the `<coderef>` element causes the target code to be displayed inline. If the target code contains non-XML characters such as '<' or '&', those characters need to be handled so that they can be displayed correctly by the final rendering engine.

#### Specialization hierarchy
The `<coderef>` is specialized from `<xref>`. It is defined in the programming domain module.
Attributes
The following attributes are available on this element: Universal attribute group (159), Link-relationship attribute group (167), and @keyref (180).

Example
In the following code sample, the <coderef> element references the content of the process-dita.xsl file. In the rendered output, the XSL code will be present in a code block.

```
<example>
  <title>Processing DITA</title>
  <p>This code is an example of how to process DITA.</p>
  <codeblock>
    <coderef href="process-dita.xsl"/>
  </codeblock>
</example>
```

3.3.4.5 <option>
The <option> element describes an option that can modify a command or a configuration.

Usage information
This element is part of the programming domain.

Specialization hierarchy
The <option> is specialized from <keyword>. It is defined in the programming domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
In the following code sample, ...

```
something <option>/modifier</option>
```

3.3.4.6 <parmname>
The <parmname> element identifies the name of an API parameter.

Usage information
This element is part of the programming domain.

Specialization hierarchy
The <parmname> is specialized from <keyword>. It is defined in the programming domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).
Example

In the following code sample, the `<parmname>` element identifies a parameter used with the `config` command:

```xml
<p>Use the `<parmname>/env</parmname>` parameter of the `<cmdname>config</cmdname>` command to update the field value.</p>
```

3.3.4.7 `<parml>`

The `<parml>` element identifies a specialized definition list that is designed for documenting API parameters.

Usage information

This element is part of the programming domain.

Specialization hierarchy

The `<parml>` is specialized from `<dl>`. It is defined in the programming domain module.

Attributes

STUB CONTENT

Example

Example source:

```xml
This code example is a basic method signature:
<codeblock>
returnType methodName(pList1, pList2) {
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>
</codeblock>
```

Example output:

This code example is a basic method signature:

```java
returnType methodName(pList1, pList2) {
where

pList1
  is the first variable declaration passed to methodName

pList2
  is the second variable declaration passed to methodName
```
3.3.4.8 `<plentry>`
The `<plentry>` element contains one or more parameter terms and definitions (`<pt>` and `<pd>`).

Usage information

Specialization hierarchy
+ topic/dlentry pr-d/plentry

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
See `parml` (114).

3.3.4.9 `<pt>`
The `<pt>` element specifies a parameter term within a parameter list entry.

Usage information

Specialization hierarchy
+ topic/dt pr-d/pt

Attributes
The following attributes are available on this element: Universal attribute group (159) and `@keyref` (180).

Example
See `parml` (114).

3.3.4.10 `<pd>`
The `<pd>` element specifies a parameter definition within a parameter list entry.

Usage information

Specialization hierarchy
+ topic/dd pr-d/pd

Attributes
The following attributes are available on this element: Universal attribute group (159).
Example
See parml (114).

3.3.4.11 <synph>
The <synph> element identifies a syntax phrase.

Usage information
The <synph> element is used when a complete syntax diagram is not needed, but some of the syntax elements, such as <kwd>, <oper>, or <delim> are used within the text flow of the topic content.

This element is part of the programming domain.

Specialization hierarchy
The <synph> is specialized from <ph>. It is defined in the programming domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
In the following code sample, the <synph> element ...

```xml
<synph><kwd>format</kwd> <var>volumename</var></synph>
```

3.3.4.12 <syntaxdiagram>
The <syntaxdiagram> element represents the syntax of a command, function call, or programming language statement.

Usage information
This element is part of the programming domain.

Specialization hierarchy
The <syntaxdiagram> is specialized from <fig>. It is defined in the programming domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and Display attribute group (166).

Example
In the following code sample, ...

```xml
<syntaxdiagram>
<title>CopyFile</title>
<groupseq><kwd>COPYF</kwd></groupseq>
<groupcomp><var>input-filename</var><kwd>*INFILE</kwd></groupcomp>
<groupseq><var>output-filename</var><kwd>*OUTFILE</kwd></groupseq>
<groupchoice> <var>input-filename</var> <kwd>*INFILE</kwd></groupchoice>
```
### 3.3.4.13 <groupseq>

The `<groupseq>` element is part of the subset of elements that define syntax diagrams in DITA.

**Usage information**

A `groupseq` is a logical set of pieces of syntax that go together. Within the syntax definition, groups of keywords, delimiters and other syntax units act as a combined unit, and they occur in a specific sequence, as delimited by the `<groupseq>` element.

This element is part of the programming domain.

**Specialization hierarchy**

The `<groupseq>` element is specialized from `<figgroup>`. It is defined in the programming domain module.

**Attributes**

STUB CONTENT

**Example**

In the following code sample:

```xml
<syntaxdiagram frame="bottom">
  <title>CopyFile</title>
  <groupseq><kwd>COPYF</kwd></groupseq>
  <groupcomp><var>input-filename</var><kwd>*INFILE</kwd></groupcomp>
  <groupseq><var>output-filename</var><kwd>*OUTFILE</kwd></groupseq>
  <groupchoice><var>input-filename</var><kwd>*INFILE</kwd></groupchoice>
  <groupchoice><var>output-filename</var><kwd>*OUTFILE</kwd></groupchoice>
</syntaxdiagram>
```

### 3.3.4.14 <groupchoice>

The `<groupchoice>` element is part of the subset of elements that define syntax diagrams in DITA.

**Usage information**

A group is a logical set of pieces of syntax that go together. A group choice specifies that the user must make a choice about which part of the syntax to use. Groups are often nested.

This element is part of the programming domain.

**Specialization hierarchy**

The `<groupchoice>` element is specialized from `<figgroup>`. It is defined in the programming domain module.

**Attributes**

STUB CONTENT
3.3.4.15 <groupcomp>

The <groupcomp> element is part of the subset of elements that define syntax diagrams in DITA.

Usage information

A group is a logical set of pieces of syntax that go together. The group composite means that the items that make up the syntax diagram will be formatted close together rather than being separated by a horizontal or vertical line, which is the usual formatting method.

Specialization hierarchy

+ topic/figgroup pr-d/groupcomp

Attributes

STUB CONTENT

Example

```xml
<syntaxdiagram frame="bottom">
  <title>CopyFile</title>
  <groupseq><kwd>COPYF</kwd></groupseq>
  <groupcomp><var>input-filename</var><kwd>*INFILE</kwd></groupcomp>
  <groupseq><var>output-filename</var><kwd>*OUTFILE</kwd></groupseq>
  <groupchoice><var>input-filename</var><kwd>*INFILE</kwd></groupchoice>
  <groupchoice><var>output-filename</var><kwd>*OUTFILE</kwd></groupchoice>
</syntaxdiagram>
```

3.3.4.16 <fragment>

The <fragment> element contains a labeled subpart of the syntax within a <syntaxdiagram>.

Usage information

The <fragment> element allows breaking out logical chunks of a large syntax diagram into named fragments.

This element is part of the programming domain.

Specialization hierarchy

The <fragment> element is specialized from <figgroup>. It is defined in the programming domain module.

Attributes

The following attributes are available on this element: Universal attribute group (159).
Example

In the following code sample, the <fragment> element

```xml
<syntaxdiagram frame="none">
  <title>CopyFile</title>
  <groupseq><kwdd>COYF</kwdd></groupseq>
  <groupcomp><var>input-filename</var><kwdd>*INFILE</kwdd></groupcomp>
  <groupseq><var>output-filename</var><kwdd>*OUTFILE</kwdd></groupseq>
  <groupchoice><var>input-filename</var><kwdd>*INFILE</kwdd></groupchoice>
  <groupchoice><var>output-filename</var><kwdd>*OUTFILE</kwdd></groupchoice>
  <groupchoice><kwdd>*OVERLAP</kwdd><kwdd>*Prompt</kwdd></groupchoice>
</syntaxdiagram>
```

3.3.4.17 <fragref>

The <fragref> element provides a logical reference to a <fragment> element within a syntax diagram, so that you can reference a syntax fragment multiple times or pull a large section of syntax out of line for easier reading.

Usage information

Specialization hierarchy

+ topic/xref pr-d/fragref

Attributes

The following attributes are available on this element: STUB CONTENT (with a narrowed definition of @importance, given below), #unique_24/unique_24_Connect_42_outputclass, and the attributes defined below.

@href

A reference to a syntax diagram <fragment> element. The referenced <fragment> must be in the same diagram as the <fragref> element. See STUB CONTENT for detailed information on supported values and processing implications.

STUB CONTENT

STUB CONTENT

Example

This markup example:

```xml
<syntaxdiagram frame="none">
  <title>CopyFile</title>
  <groupseq><kwdd>COYF</kwdd></groupseq>
  <groupcomp><var>input-filename</var><kwdd>*INFILE</kwdd></groupcomp>
  <groupseq><var>output-filename</var><kwdd>*OUTFILE</kwdd></groupseq>
  <fragref href="#syntax/overlay"></fragref>
  <groupchoice><var>input-filename</var><kwdd>*INFILE</kwdd></groupchoice>
  <groupchoice><var>output-filename</var><kwdd>*OUTFILE</kwdd></groupchoice>
  <fragment id="overlay">
    <title>Overlay</title>
    <groupchoice><kwdd>*OVERLAP</kwdd><kwdd>*Prompt</kwdd></groupchoice>
  </fragment>
</syntaxdiagram>
```
might produce output like the following:

```
CopyFile
>>>-COPYF=*input-filename=INFILE=*output-filename=*OUTFILE------>
>>>| Overlay |---=*input-filename++++*output-filename---------<
   '=*INFILE--------' '=*OUTFILE--------'
Overlay
|---=*OVERLAP=+------------------------------------------|
   '=*Prompt='
```

3.3.4.18 <synblk>
The `<synblk>` (syntax block) element organizes small pieces of a syntax definition into a larger piece.

Usage information

Specialization hierarchy
+ topic/figgroup pr-d/synblk

Attributes

The following attributes are available on this element: Universal attribute group (159).

Example

```
<synblk>
<groupseq><kwd>this</kwd><sep>-</sep><kwd>is</kwd><sep>-</sep><var>test</var></groupseq>
</synblk>
```

3.3.4.19 <synnote>
The `<synnote>` element contains a note (similar to a footnote) within a syntax definition group or fragment.

Usage information

The syntax note explains aspects of the syntax that cannot be expressed in the markup itself. The note will appear at the bottom of the syntax diagram instead of at the bottom of the page.

Specialization hierarchy
+ topic/fn pr-d/synnote

Attributes

STUB CONTENT

Example

```
<groupcomp><var>one</var><var>two</var><var>three</var></groupcomp>
<synnote>My first syntax note.</synnote>
```
3.3.4.20 <synnoteref>
The <synnoteref> element references a syntax note element (<synnote>) that has already been defined elsewhere in the syntax diagram.

Usage information
The same notation can be used in more than one syntax definition.

Specialization hierarchy
+ topic/xref pr-d/synnoteref

Attributes
The following attributes are available on this element: STUB CONTENT, #unique_24/unique_24_Connect_42_outputclass, and the attribute defined below.

@href
A reference to the target syntax note (<synnote>) element. The referenced syntax note must be in the same syntax diagram as the <synnoteref> element. See STUB CONTENT for detailed information on supported values and processing implications.

Example

```
<synnoteref href="#topicid/mysyn"/>
```

3.3.4.21 <kwds>
The <kwds> element identifies a keyword within a syntax diagram or phrase.

Usage information
A keyword is typed or output, by the user or application, exactly as specified in the syntax diagram or phrase.

This element is part of the programming domain.

Specialization hierarchy
The <kwds> is specialized from <keyword>. It is defined in the programming domain module.

Attributes
STUB CONTENT

Example
In the following code sample, the <kwds> element identifies text that must be provided to the application exactly as specified:

```xml
<syntaxdiagram frame="bottom">
  <title>CopyFile</title>
  <groupseq>
    <kwds>COPYF</kwds>
  </groupseq>
  <groupcomp><var>input-filename</var><kwds>*INFILE</kwds></groupcomp>
  <groupseq><var>output-filename</var><kwds>*OUTFILE</kwds></groupseq>
  <groupchoice><var>input-filename</var><kwds>*INFILE</kwds></groupchoice>
```
3.3.4.22 <var>
The <var> element identifies a variable within a syntax diagram or phrase.

Usage information
This element is part of the programming domain.

Specialization hierarchy
The <var> is specialized from <ph>. It is defined in the programming domain module.

Attributes
STUB CONTENT

Example
In the following code sample, the <var> element identifies variables for which the user will substitute the names of the input and output files:

```xml
<title>CopyFile</title>
<groupseq><kwd>COPYF</kwd></groupseq>
<groupcomp><var>input-filename</var><kwd>*INFILE</kwd></groupcomp>
<groupseq><var>output-filename</var><kwd>*OUTFILE</kwd></groupseq>
<groupchoice><var>input-filename</var><kwd>*INFILE</kwd></groupchoice>
<groupchoice><var>output-filename</var><kwd>*OUTFILE</kwd></groupchoice>
</syntaxdiagram>
```

3.3.4.23 <oper>
The <oper> element identifies an operator within a syntax definition.

Usage information
Typical operators are equals (=), plus (+) or multiply (*).
This element is part of the programming domain.

Specialization hierarchy
The <oper> is specialized from <ph>. It is defined in the programming domain module.

Attributes
STUB CONTENT

Example
In the following code sample, the <oper> element specifies that the operator is plus (+):

```xml
<title>Adding</title>
<groupseq><kwd>1</kwd><oper>+</oper><var>two</var>
<delim>=</delim><kwd>something</kwd>
</syntaxdiagram>
```
3.3.4.24 <delim>
The <delim> element identifies a character that marks the beginning or end of a section within a syntax diagram.

**Usage information**
Typical delimiter characters are the parenthesis, comma, tab, vertical bar or other special characters. This element is part of the programming domain.

**Specialization hierarchy**
The <delim> is specialized from <ph>. It is defined in the programming domain module.

**Attributes**
STUB CONTENT

**Example**
In the following code sample, the <delim> element specifies that the equal sign (=) is used to mark the end of the group sequence.

```xml
<syntaxdiagram>
<title>Adding</title>
<groupseq><kwd>1</kwd><oper>+</oper><var>two</var><delim>=</delim><kwd>something</kwd></groupseq>
</syntaxdiagram>
```

3.3.4.25 <sep>
The <sep> element defines a separator character that is inline with the content of a syntax diagram.

**Usage information**
The separator occurs between keywords, operators or groups in a syntax definition. This element is part of the programming domain.

**Specialization hierarchy**
The <sep> is specialized from <ph>. It is defined in the programming domain module.

**Attributes**
STUB CONTENT

**Example**
In the following code sample, the <sep> element
3.3.4.26 <repsep>
The `<repsep>` element identifies a character used in a syntax diagram to indicate that a group of syntax elements can (or should) be repeated.

**Usage information**
If the `<repsep>` element contains a separator character such as a plus (+), this indicates that the character must be used between repetitions of the syntax elements.

This element is part of the programming domain.

**Specialization hierarchy**
The `<repsep>` is specialized from `<ph>`. It is defined in the programming domain module.

**Attributes**
STUB CONTENT

**Example**
In this example, the group can be repeated. When repeated, a comma should be used between selections.

```xml
<groupchoice>
  <repsep>,</repsep>
  <kwd>This</kwd>
  <kwd>That</kwd>
  <kwd>The other</kwd>
</groupchoice>
```

3.3.5 Release management domain
The release-management domain elements contain human-authored information about the changes that have been made to a DITA topic or map. A processor can retrieve this information and use it to assemble documents or topics that contain release note information.

3.3.5.1 `<change-completed>`
Use the `<change-completed>` element to indicate the date on which the change was completed.

**Usage information**
STUB CONTENT

**Specialization hierarchy**
- topic/data relmgmt-d/change-completed

**Attributes**
STUB CONTENT
3.3.5.2 <change-historylist>
Use the <change-historylist> element as a container for individual release notes, as represented by the <change-item> element.

Usage information
This element appears in the topic prolog or the map <topicmeta> element.

Specialization hierarchy
- topic/metadata relmgmt-d/change-historylist

Attributes
The following attributes are available on this element: STUB CONTENT, #unique_24/unique_24_Connect_42_outputclass, and #unique_24/unique_24_Connect_42_mapkeyref.

Example
This example shows three simple release notes added to a single topic. This topic is used in documentation for two products, A and B.

```xml
<prolog>
<!-- ... -->
<change-historylist>

<change-item product="productA productB">
<change-person>Tom Cihak</change-person>
<change-organization>JEDEC</change-organization>
<change-completed>2013-03-23</change-completed>
<change-summary>Made change 1 to both products</change-summary>
<data>Details of change 1</data>
</change-item>

<change-item product="productA">
<change-person>Tom Cihak</change-person>
<change-completed>2013-06-07</change-completed>
<change-summary>Made change 2 to product A</change-summary>
<data>Details of change 2</data>
</change-item>

<change-item product="productA productB">
<change-person>Tom Cihak</change-person>
<change-completed>2013-07-20</change-completed>
<change-summary>Made change 3 to both products</change-summary>
<data>Details of change 3</data>
</change-item>
</change-historylist>
<!-- ... -->
</prolog>
```

3.3.5.3 <change-item>
Use the <change-item> element to represent a record of a single change to a DITA topic or map.

Specialization hierarchy
- topic/data relmgmt-d/change-item

Attributes
STUB CONTENT
### 3.3.5.4 `<change-organization>`
Use the `<change-organization>` element to specify the name of an organization that required the change.

**Specialization hierarchy**
- topic/data relmgmt-d/change-organization

**Attributes**
- STUB CONTENT
- STUB CONTENT

### 3.3.5.5 `<change-person>`
Use the `<change-person>` element to specify the name of the person who made the change.

**Specialization hierarchy**
- topic/data relmgmt-d/change-person

**Attributes**
- STUB CONTENT
- STUB CONTENT

### 3.3.5.6 `<change-request-id>`
Use the `<change-request-id>` element to specify an identifier associated with the change request, such as an issue ID or ticket number.

**Specialization hierarchy**
- topic/data relmgmt-d/change-request-id

**Attributes**
- STUB CONTENT
- STUB CONTENT

### 3.3.5.7 `<change-request-reference>`
Use the `<change-request-reference>` element to provide information that links the change to an external tracking system.

**Specialization hierarchy**
- topic/metadata relmgmt-d/change-request-reference

**Attributes**
- STUB CONTENT
3.3.5.8 <change-request-system>
Use the `<change-request-system>` element to specify the name of an information system that manages or serves the referenced change request, for example, an issue tracking system.

Specialization hierarchy
- topic/data relmgmt-d/change-request-system

Attributes
STUB CONTENT
STUB CONTENT

3.3.5.9 <change-revisionid>
Use the `<change-revisionid>` element to specify a revision ID string that can identify the change.

Specialization hierarchy
- topic/data relmgmt-d/change-revisionid

Attributes
STUB CONTENT
STUB CONTENT

3.3.5.10 <change-started>
Use the `<change-started>` element to specify the date on which the change was initiated.

Usage information
STUB CONTENT

Specialization hierarchy
- topic/data relmgmt-d/change-started

Attributes
STUB CONTENT
STUB CONTENT

3.3.5.11 <change-summary>
Use the `<change-summary>` element to summarize the change.

Usage information
This element contains the portion of the release note that might appear in a document.
3.3.6 Software domain
The software domain elements are used to describe the operation of a software program.

3.3.6.1 <msgph>
The <msgph> element identifies the text of a message that is produced by an application or program.

Usage information
This element is part of the software domain.

Specialization hierarchy
The <msgph> element is specialized from <ph>. It is defined in the software domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
In the following code sample, the <msgph> element is used to tag a message returned by the server:

```xml
<p>A server log entry of <msgnum>I:0</msgnum> is equivalent to the text message, <msgph>informational: successful</msgph>.</p>
```

3.3.6.2 <msgblock>
The <msgblock> element contains a multi-line message or set of messages.

Usage information
The <msgblock> element can contain multiple message numbers and message descriptions, each enclosed in <msgnum> and <msgph> elements. It also can contain the message content directly.

This element is part of the software domain.

Rendering expectations
Processors SHOULD preserve line breaks and spaces that are present in the content of a <msgblock> element.

Specialization hierarchy
The <msgblock> element is specialized from <pre>. It is defined in the software domain module.
Attributes

The following attributes are available on this element: **Universal attribute group** (159), **Display attribute group** (166), **xml:space** (168), and **spectitle** (169).

Example

In the following code sample, the `<msgblock>` element contains a multi-line message returned by an application:

```xml
<p>A sequence of failed password attempts generates the following message stream:</p>
<msgblock>
I:0
S:3
I:1
S:3
I:1
S:4
S:99 (lockup)
</msgblock>
```

3.3.6.3 `<msgnum>`

The `<msgnum>` element identifies the number of a message that is produced by an application or program.

Usage information

This element is part of the software domain.

Specialization hierarchy

The `<msgnum>` element is specialized from `<keyword>`. It is defined in the software domain module.

Attributes

The following attributes are available on this element: **Universal attribute group** (159) and **@keyref** (180).

Example

In the following code sample, the `<msgnum>` element identifies the number of the message returned by an application.

```xml
<p>A server log entry of <msgnum>I:0</msgnum> is equivalent to the text message <msgph>informational: successful</msgph>.</p>
```

3.3.6.4 `<cmdname>`

The `<cmdname>` element identifies the name of a software command.

Usage information

This element is part of the software domain.

Specialization hierarchy

The `<cmdname>` element is specialized from `<keyword>`. It is defined in the software domain module.
Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
In the following code sample, the <cmdname> element identifies the name of the rm command.

<p>Use the <cmdname>rm</cmdname> command to permanently delete an object.</p>

3.3.6.5 <varname>
The <varname> element identifies a variable that is supplied to a software application.

Usage information
This element is part of the software domain.

Specialization hierarchy
The <varname> element is specialized from <keyword>. It is defined in the software domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
In the following code sample, the <varname> element is used to identify variables that represent the "installation directory," "project directory," and "filename".

<filepath>
  <varname>install-dir</varname>/projects\working\<varname>project-dir</varname>\source\<varname>filename</varname>.java
</filepath>

3.3.6.6 <filepath>
The <filepath> element identifies file names and system paths.

Usage information
This element is part of the software domain.

Specialization hierarchy
The <filepath> element is specialized from <ph>. It is defined in the software domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).
Example
In the following code sample, the `<filepath>` element is used to tag both file names and system paths:

```xml
<p>Uncompress the `<filepath>gbbrsh.gz</filepath>` file to the `<filepath>/usr</filepath>` directory. Ensure that the `<filepath>/usr/tools/data.cfg</filepath>` path is listed in the execution path system variable.&lt;/p&gt;
```

3.3.6.7 `<userinput>`
The `<userinput>` element identifies text that a user types in response to an application or system prompt.

Usage information
This element is part of the software domain.

Specialization hierarchy
The `<userinput>` element is specialized from `<ph>`. It is defined in the software domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
In the following code sample, the `<userinput>` element identifies text that a user should type at the command prompt:

```xml
<p>From a DOS command prompt, type `<userinput>dir</userinput>` to view a list of files in the current directory.&lt;/p&gt;
```

3.3.6.8 `<systemoutput>`
The `<systemoutput>` element identifies computer output or responses to a command or situation.

Usage information
This element is part of the software domain.

Specialization hierarchy
The `<systemoutput>` element is specialized from `<ph>`. It is defined in the software domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).
Example
In the following code sample, the `<systemoutput>` element identifies an application response to user input:

```xml
<p>After you type `<userinput>mealplan dinner</userinput>`, the meal planning program will print a `<systemoutput>For what day?</systemoutput>` message. Reply by typing the day of the week for which you want a meal plan, for example, `<userinput>Thursday</userinput>`.</p>
```

3.3.7 SVG domain
The SVG domain elements enable direct use of SVG markup within DITA documents, as well as use-by-reference of SVG markup that is stored in separate non-DITA documents. SVG is a W3C standard.

For SVG markup that is stored directly in DITA documents that are validated using DTDs, the SVG elements must use a namespace prefix in order to avoid conflict with DITA-defined elements of the same name. Documents validated using XSD or RELAX NG can default the SVG namespace on the SVG `<svg>` element. SVG elements that are referenced using the `<svgref>` element do not need to have a namespace prefix, because they are parsed separately from the DITA documents that refer to them. By default, the SVG domain is configured to use the namespace prefix "svg" for the SVG elements.

Related information
Scalable Vector Graphics (SVG) 1.1 (Second Edition)

3.3.7.1 `<svg-container>`
Use the `<svg-container>` element to store content that contributes to a scalable vector graphic (SVG).

Usage information
`<svg-container>` content includes SVG elements, references to SVG elements that are stored in separate, non-DITA documents, `<data>`, or `<data-about>` elements.

The SVG markup must have a root element of `<svg>` with the SVG namespace: “http://www.w3.org/2000/svg”.

Specialization hierarchy
+ topic/foreign/svg-d/svg-container

Attributes
The following attributes are available on this element: Universal attribute group (159).

Example
In the following example, an `<svg-container>` element contains inline SVG markup:

```xml
<topic id="svg-test-topic-01">
  <title>SVG Domain Test: Namespace Prefixed SVG Elements</title>
  <body>
    <p>SVG inline: <svg-container>
      <svg:svg width="100" height="100">
        <svg:defs>
          <svg:filter id="f1"
```

standards-track-work-product
Copyright © OASIS Open 2019. All Rights Reserved. 27 August 2019
Page 132 of 218
In the following example, the `<svgref>` element uses the `@keyref` attribute to address SVG markup that is stored in a separate, non-DITA document:

```xml
<fig>
  <title>Figure with SVG Container</title>
  <svg-container>
    <svgref keyref="svg-fragment-01"/>
  </svg-container>
</fig>
```

The key "svg-fragment-01" is declared in the following way:

```xml
<map>
  ...
  <keydef keys="svg-fragment-01"
    href="media/svg/svg-library.xml#frag-0001"
    format="svg"/>
</map>
```
3.3.7.2 <svgref>
Use the <svgref> element to reference a non-DITA XML document that contains scalable vector graphic (SVG) markup.

Usage information
This element enables you to use the markup by reference.

The reference must be to a SVG <svg> element. The reference can be one of the following:

- A URI that addresses an XML document; the XML document has a SVG <svg> element as the root element
- A URI that addresses an XML document and contains a fragment identifier that is the XML ID of a <svg> element within the document

Processors SHOULD process the SVG as though the <svg> element had occurred directly in the content of the containing <svg-container> element.

The reference can be direct, using the @href attribute, or indirect, using the @keyref attribute. For indirect referencing, only the key name should be specified. The ID of the <svg> element must be specified as part of the value for the @href attribute on the key definition.

For example, to refer to the <svg> element with the @id of "svg-fragment-02" within a larger document using a key reference, you would define the key in the following way:

```xml
<keydef
    keys="svg-fragment-0002"
    href="svg/svg-library.xml#svg-fragment-02"
/>
```

You would refer to this key using just the key name:

```xml
<svg-container>
    <svgref keyref="svg-fragment-0002"/>
</svg-container>
```

Specialization hierarchy
+ topic/xref svg-d/svgref

Attributes
The following attributes are available on this element: STUB CONTENT, #unique_24/unique_24_Connect_42_outputclass, and @keyref. This element also uses STUB CONTENT, with narrowed definitions of @href and @format (given below).

@href
Reference to the <svg> element to be used. If the <svg> element is the root element of the referenced resource, then no fragment identifier is required, otherwise, a fragment identifier must be specified, where the fragment identifier is the XML ID of the <svg> element to be used. This attribute is not required, but must be specified if @keyref is not specified. See STUB CONTENT for detailed information on supported values and processing implications.
@format
The data format of the resource. For SVG the format should be "svg", which is the default value for @format on this element.

Example
Here is a reference to an <svg> element that is the root element of its containing document:

```xml
<fig>
  <title>Figure With SVG Container</title>
  <svg-container>
    <svgref href="media/svg/svg-graphic-01.xml" format="svg"/>
  </svg-container>
</fig>
```

The svg-graphic-01.xml file contains the following content. Note that the <svg> element sets the SVG namespace as the default namespace, so there are no namespace prefixes on the SVG markup:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg" width="100" height="100">
  <defs>
    <filter id="f1" x="0" y="0">
      <feGaussianBlur in="SourceGraphic" stdDeviation="15"/>
    </filter>
  </defs>
  <rect width="90" height="90" stroke="green" stroke-width="3" fill="yellow" filter="url(#f1)"/>
</svg>
```

Here is a reference to a specific <svg> element in a containing XML file:

```xml
<fig>
  <title>Figure with SVG Container</title>
  <svg-container>
    <svgref href="media/svg/svg-library.xml#frag-0001"/>
  </svg-container>
</fig>
```

The svg-library.xml file contains the following content:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <part>
    <svg id="frag-0001" xmlns="http://www.w3.org/2000/svg" width="100" height="100">
      <defs>
        <filter id="f1" x="0"/>
      </defs>
    </svg>
  </part>
</root>
```
3.3.8 User interface domain

The user interface domain elements are used to describe the user interface of a software program.

3.3.8.1 <uicontrol>

The <uicontrol> element identifies user interface controls, such as names of buttons, fields, menu items, and other objects that enable users to control the interface.

Usage information

The <uicontrol> element also is used inside a <menucascade> element to identify a sequence of menu choices in a nested menu, such as File > New.

This element is part of the user interface domain.

Specialization hierarchy

The <uicontrol> element is specialized from <ph>. It is defined in the user-interface domain module.

Attributes

The following attributes are available on this element: Universal attribute group (159) and @keyref (180).
Example
In the following code sample, the `<uicontrol>` element identifies a button that a user is directed to press:

```xml
<p>Press <uicontrol>OK</uicontrol>.</p>
```

3.3.8.2 `<wintitle>`
The `<wintitle>` element identifies named windows and dialogs.

Usage information
This element is part of the user interface domain.

Specialization hierarchy
The `<wintitle>` element is specialized from `<keyword>`. It is defined in the user-interface domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and `@keyref` (180).

Example
In the following code sample, the `<wintitle>` element is used to tag the name of the "Configuration Options” window:

```xml
<step><cmd>Click <uicontrol>Configure</uicontrol>.</cmd>
<stepresult>The `<wintitle>Configuration Options</wintitle>` window opens with your last set of selections highlighted.</stepresult>
</step>
```

3.3.8.3 `<menucascade>`
The `<menucascade>` element identifies a sequence of menu choices in a nested menu, such as File > New.

Usage information
This element is part of the user interface domain.

Specialization hierarchy
The `<menucascade>` element is specialized from `<ph>`. It is defined in the user-interface domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and `@keyref` (180).
**Example**

In the following code sample, the `<menucascade>` element is used to identify a series of menu choices that enable users to launch the Notepad application:

```xml
<menucascade>
  <uicontrol>Start</uicontrol>
  <uicontrol>Programs</uicontrol>
  <uicontrol>Accessories</uicontrol>
  <uicontrol>Notepad</uicontrol>
</menucascade>
```

### 3.3.8.4 `<shortcut>`

The `<shortcut>` element identifies a keyboard shortcut for a menu or window action.

**Usage information**

This element is part of the user interface domain.

**Specialization hierarchy**

The `<shortcut>` element is specialized from `<keyword>`. It is defined in the user-interface domain module.

**Attributes**

The following attributes are available on this element: **Universal attribute group** (159) and `@keyref` (180).

**Example**

In the following code sample, the `<shortcut>` element identifies the keyboard shortcut for the "Start Programs" menu action:

```xml
<menucascade>
  <uicontrol>Start</uicontrol>
  <uicontrol><shortcut>P</shortcut>rogram</uicontrol>
</menucascade>
```

### 3.3.8.5 `<screen>`

The `<screen>` element contains a textual representation of a terminal console or other text-based computer interfaces.

**Usage information**

This element is part of the user interface domain.

**Specialization hierarchy**

The `<screen>` element is specialized from `<pre>`. It is defined in the user-interface domain module.

**Attributes**

STUB CONTENT
Example

In the following code sample, the `<screen>` element is used to provide a representation of a DOS window:

```xml
<screen>
  File  Edit  Search  View  Options  Help
  +--------------------------------- UNTITLED1 ----------------------------------+
  ¦                                                                              ¦
  ¦                                                                              ¦
  ¦                                                                              ¦
  ¦                                                                              ¦
  ¦ Line:1    Col:1  F1=Help                                                    ¦
  +------------------------------------------------------------------------------+
</screen>
```

3.3.9 XML mention domain

The XML mention domain is designed to describe and document XML document type and applications. It also can enable typographic styling, search and retrieval, and automatic indexing for XML constructs.

**Note** Although the original XML 1.0 Recommendation reserved element names beginning with "xml" or "XML" for the use of the XML standard itself, the subsequent introduction of namespaces made the restriction unnecessary. The restriction was formally removed in the XML 1.0 Fifth Edition Specification Errata. The OASIS DITA Technical Committee acknowledges this revised policy in its use of the prefix "xml" for the XML mention domain.

3.3.9.1 `<numcharref>`

The `<numcharref>` element identifies mentions of XML numeric character references.

**Usage information**

The content of the `<numcharref>` element should be the numeric value without any leading or trailing characters, for example, 10 or x0a.

This element is part of the XML mention domain.

**Specialization hierarchy**

The `<numcharref>` element is specialized from `<markupname>`; it is defined in the XML-mention domain module. The `<markupname>` element is specialized from `<keyword>`; it is defined in the markup-name domain module.

**Attributes**

The following attributes are available on this element: Universal attribute group (159) and `@keyref` (180).

**Example**

In the following code sample, a `<numcharref>` element tags the numeric character reference for the a-acute Unicode character:

```xml
<p>Numeric character references represent characters from the Universal Character Set (UCS) of Unicode. They are used to reference characters that cannot easily be directly encoded in a document, such as a copyright symbol. When a markup-aware processor encounters a numeric character reference, for example, `<numcharref>225</numcharref>`, it renders the reference as the Unicode character that it represents: a-acute.</p>
```
3.3.9.2 <parameterentity>
The <parameterentity> element identifies mentions of parameter entities.

Usage information
The content of the <parameterentity> element should be the entity name without a leading percentage sign or trailing semi-colon, for example, keyword.content.
This element is part of the XML mention domain.

Specialization hierarchy
The <parameterentity> element is specialized from <markupname>; it is defined in the XML-mention domain module. The <markupname> element is specialized from <keyword>; it is defined in the markup-name domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Examples
In the following code sample, the <parameterentity> element tags the name of the %xml-d-dec; parameter entity:

```xml
<p>To include the XML-mention domain in a DTD document-type shell, you must declare the <parameterentity>xml-d-dec</parameterentity> parameter entity.</p>
```

3.3.9.3 <textentity>
The <textentity> element identifies mentions of XML text entities.

Usage information
The content of the <textentity> element should be the entity name without the ampersand and semi-colon delimiters, for example, hi-d-att.
This element is part of the XML mention domain.

Specialization hierarchy
The <textentity> element is specialized from <markupname>; it is defined in the XML-mention domain module. The <markupname> element is specialized from <keyword>; it is defined in the markup-name domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).
Example
In the following code sample, the `<textentity>` element is used to tag the token that the highlighting domain contributes to the `@domains` attribute:

```
The <textentity>hi-d-att</textentity> entity holds the contribution for the `<xmlatt>domains</xmlatt>` attribute.
```

3.3.9.4 `<xmlatt>`
The `<xmlatt>` element identifies mentions of XML attributes.

Usage information
The content of the `<xmlatt>` element should be the attribute name without commercial at symbol (@) or equals character (=), for example, audience.

This element is part of the XML mention domain.

Specialization hierarchy
The `<xmlatt>` element is specialized from `<markupname>`; it is defined in the XML-mention domain module. The `<markupname>` element is specialized from `<keyword>`; it is defined in the markup-name domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and `@keyref` (180).

Example
In the following code sample, the `<xmlatt>` element is used to tag mentions of the `@collection-type` and `@linking` attributes:

```
The `<xmlatt>collection-type</xmlatt>` and `<xmlatt>linking</xmlatt>` attributes affect how related links are generated for topics that are referenced in the DITA map.
```

3.3.9.5 `<xmlelement>`
The `<xmlelement>` element identifies mentions of XML element types.

Usage information
The content of the `<xmlelement>` element should be the element type name without leading or trailing angle brackets.

This element is part of the XML mention domain.

Specialization hierarchy
The `<xmlelement>` element is specialized from `<markupname>`; it is defined in the XML-mention domain module. The `<markupname>` element is specialized from `<keyword>`; it is defined in the markup-name domain module.
Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
In the following code sample, the `<xmlelement>` element tags the `<uicontrol>` element from the user-interface domain.

```xml
<p>Use the `<xmlelement>uicontrol</xmlelement>` (user interface control) element to indicate the names of buttons, entry fields, menu items, or other objects that enable a user to interact with a graphical user interface.</p>
```

3.3.9.6 `<xmlnsname>`
The `<xmlnsname>` element identifies mentions of namespace names.

Usage information
This element is part of the XML mention domain.

Specialization hierarchy
The `<xmlnsname>` element is specialized from `<markupname>`; it is defined in the XML-mention domain module. The `<markupname>` element is specialized from `<keyword>`; it is defined in the markup-name domain module.

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
In the following code sample, an `<xmlnsname>` element tags the namespace for XHTML:

```xml
The namespace name for XHTML is `<xmlnsname>http://www.w3.org/1999/xhtml</xmlnsname>`.
```

3.3.9.7 `<xmlpi>`
The `<xmlpi>` element identifies mentions of processing instruction names.

Usage information
This element is part of the XML mention domain.

Specialization hierarchy
The `<xmlpi>` element is specialized from `<markupname>`; it is defined in the XML-mention domain module. The `<markupname>` element is specialized from `<keyword>`; it is defined in the markup-name domain module.
**Attributes**

The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

**Example**

In the following code sample, an `<xmlpi>` element is used to tag the name of a processing instruction:

```
While DITA does not define any processing instructions, some applications might use some DocBook processing instructions, such as `<xmlpi>dbhtmlbgcolor</xmlpi>`.  
```

### 3.3.10 xNAL domain

The xNAL domain elements represent a subset of the Extensible Name and Address Standard. It is used to encode information about the author or authors of DITA information. The domain can be included in any DITA document type shell that requires additional metadata for names and addresses, although the implementations provided by OASIS only include it in the bookmap document type.

#### 3.3.10.1 `<authorinformation>`

The `<authorinformation>` element contains detailed information about the author or authoring organization.

**Specialization hierarchy**

+ topic/author xnal-d/authorinformation

**Attributes**

STUB CONTENT

**Example**

```
<authorinformation>
  <personinfo>
    <namedetails><personname>
      <firstname>Derek</firstname>
      <middlename>L.</middlename>
      <lastname>Singleton</lastname>
      <generationidentifier>Jr.</generationidentifier>
      <otherinfo>noted psychologist</otherinfo>
    </personname></namedetails>
    <addressdetails>
      <thoroughfare>123 Yellow Brick Road</thoroughfare>
      <locality>Emerald City</locality>
      <administrativearea>Kansas</administrativearea>
      <country>USA</country>
    </addressdetails>
    <contactnumbers><contactnumber>123-555-4678</contactnumber></contactnumbers>
    <emailaddresses><emailaddress>wizard@example.org</emailaddress></emailaddresses>
  </personinfo>
</authorinformation>
```
3.3.10.2 <addressdetails>
The <addressdetails> element contains information about the address of the author or authoring group.

Specialization hierarchy
+ topic/ph xnal-d/addressdetails

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example

```xml
<personinfo>
  <namedetails>
    <firstname>Derek</firstname>
    <middlename>L.</middlename>
    <lastname>Singleton</lastname>
    <generationidentifier>Jr.</generationidentifier>
    <otherinfo>noted psychologist</otherinfo>
  </namedetails>
  <addressdetails>
    <thoroughfare>123 Yellow Brick Road</thoroughfare>
    <locality>Emerald City</locality>
    <administrativearea>Kansas</administrativearea>
    <country>USA</country>
  </addressdetails>
  <contactnumbers>
    <contactnumber>123-555-4678</contactnumber>
  </contactnumbers>
  <emailaddresses>
    <emailaddress>wizard@example.org</emailaddress>
  </emailaddresses>
</personinfo>
```

3.3.10.3 <administrativearea>
The <administrativearea> element contains information about a county, state, or province.

Specialization hierarchy
+ topic/ph xnal-d/administrativearea

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example

```
<addressdetails>
  <thoroughfare>123 Yellow Brick Road</thoroughfare>
  <locality>Emerald City</locality>
  <administrativearea>Kansas</administrativearea>
  <country>USA</country>
</addressdetails>
```
3.3.10.4 <contactnumber>
A <contactnumber> element contains the contact number of a person or organization, such as a telephone number, mobile phone number, or fax number.

Specialization hierarchy
+ topic/data xnal-d/contactnumber

Attributes
STUB CONTENT

Example

```xml
<personinfo>
  <namedetails><personname>
    <firstname>Derek</firstname>
    <middlename>L.</middlename>
    <lastname>Singleton</lastname>
    <generationidentifier>Jr.</generationidentifier>
    <otherinfo>noted psychologist</otherinfo>
  </personname></namedetails>
  <addressdetails>
    <thoroughfare>123 Yellow Brick Road</thoroughfare>
    <locality>Emerald City</locality>
    <administrativearea>Kansas</administrativearea>
    <country>USA</country>
  </addressdetails>
  <contactnumbers><contactnumber>123-555-4678</contactnumber></contactnumbers>
</personinfo>
```

3.3.10.5 <contactnumbers>
The <contactnumbers> element contains a list of telephone and fax numbers.

Specialization hierarchy
+ topic/data xnal-d/contactnumbers

Attributes
STUB CONTENT

Example

```xml
<personinfo>
  <namedetails><personname>
    <firstname>Derek</firstname>
    <middlename>L.</middlename>
    <lastname>Singleton</lastname>
    <generationidentifier>Jr.</generationidentifier>
    <otherinfo>noted psychologist</otherinfo>
  </personname></namedetails>
  <addressdetails>
    <thoroughfare>123 Yellow Brick Road</thoroughfare>
    <locality>Emerald City</locality>
    <administrativearea>Kansas</administrativearea>
    <country>USA</country>
  </addressdetails>
  <contactnumbers><contactnumber>123-555-4678</contactnumber></contactnumbers>
</personinfo>
```
3.3.10.6 <country>
The `<country>` element contains the name of a country.

**Specialization hierarchy**
+ topic/ph xnal-d/country

**Attributes**
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

**Example**
```
<addressdetails>
    <thoroughfare>123 Yellow Brick Road</thoroughfare>
    <locality>Emerald City</locality>
    <administrativearea>Kansas</administrativearea>
    <country>USA</country>
</addressdetails>
```

3.3.10.7 <emailaddress>
The `<emailaddress>` element contains an e-mail address.

**Specialization hierarchy**
+ topic/data xnal-d/emailaddress

**Attributes**
STUB CONTENT

**Example**
```
<personinfo>
    <namedetails>
        <personname>
            <firstname>Derek</firstname>
            <middlename>L.</middlename>
            <lastname>Singleton</lastname>
            <generationidentifier>Jr.</generationidentifier>
            <otherinfo>noted psychologist</otherinfo>
        </personname>
    </namedetails>
    <addressdetails>
        <thoroughfare>123 Yellow Brick Road</thoroughfare>
        <locality>Emerald City</locality>
        <administrativearea>Kansas</administrativearea>
        <country>USA</country>
    </addressdetails>
    <contactnumbers>
        <contactnumber>123-555-4678</contactnumber>
    </contactnumbers>
    <emailaddresses>
        <emailaddress>wizard@example.org</emailaddress>
    </emailaddresses>
</personinfo>
```
3.3.10.8 <emailaddresses>
The <emailaddresses> element contains a list of e-mail addresses.

Specialization hierarchy
+ topic/data xnal-d/emailaddresses

Attributes
STUB CONTENT

Example

```xml
<personinfo>
  <namedetails>
    <personname>
      <firstname>Derek</firstname>
      <middlename>L.</middlename>
      <lastname>Singleton</lastname>
      <generationidentifier>Jr.</generationidentifier>
      <otherinfo>noted psychologist</otherinfo>
    </personname>
    <addressdetails>
      <thoroughfare>123 Yellow Brick Road</thoroughfare>
      <locality>Emerald City</locality>
      <administrativearea>Kansas</administrativearea>
      <country>USA</country>
    </addressdetails>
    <contactnumbers>
      <contactnumber>123-555-4678</contactnumber>
    </contactnumbers>
    <emailaddresses>
      <emailaddress>wizard@example.org</emailaddress>
    </emailaddresses>
  </namedetails>
</personinfo>
```

3.3.10.9 <firstname>
The <firstname> element contains the person's first name.

Specialization hierarchy
+ topic/data xnal-d/firstname

Attributes
STUB CONTENT

Example

```xml
<namedetails>
  <personname>
    <honorific>Dr.</honorific>
    <firstname>Derek</firstname>
    <middlename>L.</middlename>
    <lastname>Singleton</lastname>
    <generationidentifier>Jr.</generationidentifier>
    <otherinfo>noted psychologist</otherinfo>
  </personname>
</namedetails>
```
3.3.10.10 <generationidentifier>
The <generationidentifier> element contains information about the person's generation, such as: Jr, III, or VIII.

Specialization hierarchy
+ topic/data xnal-d/generationidentifier

Attributes
STUB CONTENT

Example

```xml
<namedetails><personname>
  <firstname>Derek</firstname>
  <middlename>L.</middlename>
  <lastname>Singleton</lastname>
  <generationidentifier>Jr.</generationidentifier>
  <otherinfo>noted psychologist</otherinfo>
</personname></namedetails>
```

3.3.10.11 <honorific>
The <honorific> element contains the person's title, such as: Dr., Mr., Ms., or HRH.

Specialization hierarchy
+ topic/data xnal-d/honorific

Attributes
STUB CONTENT

Example

```xml
<namedetails><personname>
  <honorific>Dr.</honorific>
  <firstname>Derek</firstname>
  <middlename>L.</middlename>
  <lastname>Singleton</lastname>
  <generationidentifier>Jr.</generationidentifier>
  <otherinfo>noted psychologist</otherinfo>
</personname></namedetails>
```

3.3.10.12 <lastname>
The <lastname> element contains the person's last name.

Specialization hierarchy
+ topic/data xnal-d/lastname

Attributes
STUB CONTENT
3.3.10.13 <locality>
The <locality> element contains information about the city and postal or ZIP code.

Usage information
This element can contain the information directly, or by acting as a wrapper for <localityname> and <postalcode>.

Specialization hierarchy
+ topic/ph xnal-d/locality

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
<addressdetails>
<thoroughfare>123 Yellow Brick Road</thoroughfare>
<locality>
<localityname>Emerald City</localityname>
<postalcode>66780</postalcode>
</locality>
<administrativearea>Kansas</administrativearea>
<country>USA</country>
</addressdetails>

3.3.10.14 <localityname>
The <localityname> element contains the name of the locality or city.

Specialization hierarchy
+ topic/ph xnal-d/localityname

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example
<addressdetails>
<thoroughfare>123 Yellow Brick Road</thoroughfare>
<locality>
3.3.10.15 <middlename>
The <middlename> element contains the person's middle name or initial.

Specialization hierarchy
+ topic/data xnal-d/middlename

Attributes
STUB CONTENT

Example

3.3.10.16 <namedetails>
The <namedetails> element contains information about the name of the author or the authoring organization.

Specialization hierarchy
+ topic/data xnal-d/namedetails

Attributes
STUB CONTENT

Example
3.3.10.17 <organizationinfo>
The `<organizationinfo>` element contains detailed information about an authoring organization.

**Specialization hierarchy**
+ topic/data xnal-d/organizationinfo

**Attributes**
STUB CONTENT

**Example**

```
<organizationinfo>
  <namedetails>
    <organizationnamedetails>
      <organizationname>WizardWorks, Inc.</organizationname>
      <otherinfo>'Best wizard in Oz'</otherinfo>
    </organizationnamedetails>
  </namedetails>
  <addressdetails>
    <thoroughfare>123 Yellow Brick Road</thoroughfare>
    <locality>Emerald City</locality>
    <administrativearea>Kansas</administrativearea>
    <country>USA</country>
  </addressdetails>
  <contactnumbers><contactnumber>123-555-4678</contactnumber></contactnumbers>
  <emailaddresses><emailaddress>wizard@example.org</emailaddress></emailaddresses>
</organizationinfo>
```

3.3.10.18 <organizationname>
The `<organizationname>` element contains name information about the authoring organization.

**Specialization hierarchy**
+ topic/ph xnal-d/organizationname

**Attributes**
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

**Example**

```
<organizationinfo>
  <namedetails>
    <organizationnamedetails>
      <organizationname>WizardWorks, Inc.</organizationname>
      <otherinfo>'Best wizard in Oz'</otherinfo>
    </organizationnamedetails>
  </namedetails>
  <addressdetails>
    <thoroughfare>123 Yellow Brick Road</thoroughfare>
    <locality>Emerald City</locality>
    <administrativearea>Kansas</administrativearea>
    <country>USA</country>
  </addressdetails>
  <contactnumbers><contactnumber>123-555-4678</contactnumber></contactnumbers>
  <emailaddresses><emailaddress>wizard@example.org</emailaddress></emailaddresses>
</organizationinfo>
```
3.3.10.19 <organizationnamedetails>
The <organizationnamedetails> element contains information about the name of an authoring organization.

Specialization hierarchy
+ topic/ph xnal-d/organizationnamedetails

Attributes
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

Example

```xml
<organizationinfo>
  <namedetails>
    <organizationnamedetails>
      <organizationname>WizardWorks, Inc.</organizationname>
      <otherinfo>'Best wizard in Oz'</otherinfo>
    </organizationnamedetails>
  </namedetails>
  <addressdetails>
    <thoroughfare>123 Yellow Brick Road</thoroughfare>
    <locality>Emerald City</locality>
    <administrativearea>Kansas</administrativearea>
    <country>USA</country>
  </addressdetails>
  <contactnumbers>
    <contactnumber>123-555-4678</contactnumber>
  </contactnumbers>
  <emailaddresses>
    <emailaddress>wizard@example.org</emailaddress>
  </emailaddresses>
  <urls>
    <url>www.wizardworks.example.org</url>
  </urls>
</organizationinfo>
```

3.3.10.20 <otherinfo>
The <otherinfo> element contains other name information about the author or authoring organization.

Specialization hierarchy
+ topic/data xnal-d/otherinfo

Attributes
STUB CONTENT

Example

```xml
<organizationinfo>
  <namedetails>
    <organizationnamedetails>
      <organizationname>WizardWorks, Inc.</organizationname>
      <otherinfo>'Best wizard in Oz'</otherinfo>
    </organizationnamedetails>
  </namedetails>
  <addressdetails>
    <thoroughfare>123 Yellow Brick Road</thoroughfare>
    <locality>Emerald City</locality>
    <administrativearea>Kansas</administrativearea>
  </addressdetails>
</organizationinfo>
```
3.3.10.21 <personinfo>
The <personinfo> element is a wrapper containing all relevant data about a person, including name, address, and contact information.

Specialization hierarchy
+ topic/data xnal-d/personinfo

Attributes
STUB CONTENT

Example

```xml
<personinfo>
  <namedetails>
    <personname>
      <firstname>Derek</firstname>
      <middlename>L.</middlename>
      <lastname>Singleton</lastname>
      <generationidentifier>Jr.</generationidentifier>
      <otherinfo>noted psychologist</otherinfo>
    </personname>
  </namedetails>
  <addressdetails>
    <thoroughfare>123 Yellow Brick Road</thoroughfare>
    <locality>Emerald City</locality>
    <administrativearea>Kansas</administrativearea>
    <country>USA</country>
  </addressdetails>
  <contactnumbers>
    <contactnumber>123-555-4678</contactnumber>
  </contactnumbers>
  <emailaddresses>
    <emailaddress>wizard@example.org</emailaddress>
  </emailaddresses>
</personinfo>
```

3.3.10.22 <personname>
The <personname> element contains name information about the author.

Specialization hierarchy
+ topic/data xnal-d/personname

Attributes
STUB CONTENT

Example

```xml
<personinfo>
  <namedetails>
    <personname>
      <firstname>Derek</firstname>
      <middlename>L.</middlename>
      <lastname>Singleton</lastname>
      <generationidentifier>Jr.</generationidentifier>
      <otherinfo>noted psychologist</otherinfo>
    </personname>
  </namedetails>
</personinfo>
```
3.3.10.23 <postalcode>
The <postalcode> element contains information about the postal code or the ZIP code.

**Specialization hierarchy**
+ topic/ph xnal-d/postalcode

**Attributes**
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

**Example**

```xml
<addressdetails>
<thoroughfare>123 Yellow Brick Road</thoroughfare>
<locality>Emerald City</locality>
<administrativearea>Kansas</administrativearea>
<country>USA</country>
</addressdetails>
```

3.3.10.24 <thoroughfare>
The <thoroughfare> element contains information about the thoroughfare - for example, the street, avenue, or boulevard - on which an address is located.

**Specialization hierarchy**
+ topic/ph xnal-d/thoroughfare

**Attributes**
The following attributes are available on this element: Universal attribute group (159) and @keyref (180).

**Example**

```xml
<addressdetails>
<thoroughfare>123 Yellow Brick Road</thoroughfare>
<locality>Emerald City</locality>
<administrativearea>Kansas</administrativearea>
<country>USA</country>
</addressdetails>
```
3.3.10.25 <url>
The <url> element contains a Uniform Resource Locator (URL), such as a person's or company's internet address.

Specialization hierarchy
+ topic/data xnal-d/url

Attributes
STUB CONTENT

Example

```xml
<organizationinfo>
  <namedetails>
    <organizationnamedetails>
      <organizationname>WizardWorks, Inc.</organizationname>
      <otherinfo>'Best wizard in Oz'</otherinfo>
    </organizationnamedetails>
  </namedetails>
  <addressdetails>
    <thoroughfare>123 Yellow Brick Road</thoroughfare>
    <locality>Emerald City</locality>
    <administrativearea>Kansas</administrativearea>
    <country>USA</country>
  </addressdetails>
  <contactnumbers><contactnumber>123-555-4678</contactnumber></contactnumbers>
  <emailaddresses><emailaddress>wizard@example.org</emailaddress></emailaddresses>
  <urls><url>www.wizardworks.example.org</url></urls>
</organizationinfo>
```

3.3.10.26 <urls>
The <urls> element contains a list of Uniform Resource Locators (URLs).

Specialization hierarchy
+ topic/data xnal-d/urls

Attributes
STUB CONTENT

Example

```xml
<organizationinfo>
  <namedetails>
    <organizationnamedetails>
      <organizationname>WizardWorks, Inc.</organizationname>
      <otherinfo>'Best wizard in Oz'</otherinfo>
    </organizationnamedetails>
  </namedetails>
  <addressdetails>
    <thoroughfare>123 Yellow Brick Road</thoroughfare>
    <locality>Emerald City</locality>
    <administrativearea>Kansas</administrativearea>
    <country>USA</country>
  </addressdetails>
  <contactnumbers><contactnumber>123-555-4678</contactnumber></contactnumbers>
  <emailaddresses><emailaddress>wizard@example.org</emailaddress></emailaddresses>
  <urls>
    <url>www.wizardworks.example.org</url>
    <url>www.example.org</url>
  </urls>
</organizationinfo>
```
4 Conformance
Appendix A Acknowledgments

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

- Robert D. Anderson, IBM
- Bill Burns, Healthwise
- Kristen James Eberlein, Eberlein Consulting LLC
- Nancy Harrison, Individual Member
- Scott Hudson, Boeing
- Eric Sirois, IXiASOFT
- Bob Thomas, Individual Member
Appendix B Non-normative information

This section contains non-normative information, including topics about X and Y.

Appendix B.1 Attributes

This section collects commonly used attributes, with common definitions. If an element uses a different definition, or narrows the scope of, an otherwise common attribute, it will be called out in the topic that defines the element.

Appendix B.1.1 Universal attribute group

The universal attribute group defines a set of common attributes that are available on most DITA elements. The universal attribute group includes all attributes from the ID, metadata, and localization attribute groups, plus the @class attribute.

@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. The @class attribute supports specialization. Its predefined values allow DITA tools to work correctly with ranges of related content. 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 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

Names 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.

Appendix B.1.1.1 ID attribute group

The ID attribute group includes attributes that enable the naming and referencing of elements in topics and maps.

@conaction

This attribute enables users to push content into a new location. Allowable values are mark, pushafter, pushbefore, pushreplace, and -dita-use-conref-target. See The conaction attribute (169) for examples and details about the syntax.

@conkeyref

Allows the conref feature to operate using a key instead of a URI. See The conkeyref attribute (176) for more details about the syntax and behaviors.

@conref

This attribute is used to reference an ID on content that can be reused. See The conref attribute (177) for examples and details about the syntax.

@conrefend

The @conrefend attribute is used when reusing a range of elements through @conref. The syntax is the same as for the @conref attribute; see The conrefend attribute (173) for examples.

@id

An anchor point. This ID is the target for references by @href and @conref attributes and for external applications that refer to DITA 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.

Appendix B.1.1.2 Metadata attribute group

The metadata attribute group includes common metadata attributes, several of which support conditional processing (filtering and flagging) or the creation of new attribute domain specializations.

@base
A generic attribute that has no specific purpose. It is intended to act as a base for specialized attributes that have a simple value syntax like the conditional processing attributes (one or more alphanumeric values separated by whitespace), but is not itself a filtering or flagging attribute. The @base attribute takes a space-delimited set of values. However, when acting as a container for generalized attributes, the attribute values will be more complex; see Attribute generalization for more details.

@props
Root attribute from which new metadata attributes can be specialized. This is a property attribute which supports conditional processing for filtering or flagging. If no value is specified, but the attribute is specified on an ancestor within a map or within the related-links section, the value will cascade from the closest ancestor.

The @props attribute takes a space-delimited set of values. However, when acting as a container for generalized attributes, the attribute values will be more complex; see Attribute generalization for more details.

The attributes @audience, @deliveryTarget, @platform, @product, and @otherprops are specialized from the @props attribute. They are defined in independent attribute extension domains, and integrated by default into all OASIS-provided document-type shells. If any of these domains is not integrated into a given document-type shell, the relevant attribute(s) will not be available.

@audience
Indicates the intended audience for the element. If no value is specified, but the attribute is specified on an ancestor within a map or within the related-links section, the value will cascade from the closest ancestor.

@deliveryTarget
The intended delivery target of the content, for example, html, pdf, or epub. If no value is specified, but the attribute is specified on an ancestor within a map or within the related-links section, the value will cascade from the closest ancestor.

@otherprops
This attribute can be used for any other properties that might be needed to describe an audience, or to 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 an ancestor within a map or within the related-links section, the value will cascade from the closest ancestor.

@platform
Indicates operating system and hardware. If no value is specified, but the attribute is specified on an ancestor within a map or within the related-links section, the value will cascade from the closest ancestor.

@product
Contains the name of the product to which the element applies. If no value is specified, but the attribute is specified on an ancestor within a map or within the related-links section, the value will cascade from the closest ancestor.
@importance
A range of values that describe an importance or priority attributed to an element. For example, in steps of a task, the attribute indicates whether a step is optional or required. This attribute is not used for DITAVAL-based filtering or flagging; applications might use the importance value to highlight elements. Allowable values are: obsolete, deprecated, optional, default, low, normal, high, recommended, required, urgent, and -dita-use-conref-target.

@rev
Indicates 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. It is not sufficient to be used for version control. If no value is specified, but the attribute is specified on an ancestor within a map or within the related-links section, the value will cascade from the closest ancestor.

@status
The modification status of the current element. Allowable values are: new, changed, deleted, unchanged, and -dita-use-conref-target.

Appendix B.1.1.3 Localization attribute group
The localization attribute group defines a set of common attributes that are related to translation and localization. These attributes are available on most DITA elements.

@dir
Specifies the directionality of text: left-to-right (ltr, the processing default) or right-to-left (rtl). The value lro indicates an override of normal bidi text presentation, forcing the element into left-to-right mode; rlo overrides normal rules to force right-to-left presentation. Allowable values are ltr, rtl, lro, rlo, and -dita-use-conref-target. See The dir attribute for more information.

@translate
Indicates whether the content of the element should be translated or not. Allowable values are yes, no, and -dita-use-conref-target. See Element-by-element recommendations for translators for suggested processing defaults for each element.

@xml:lang
Specifies the language of the element content. The @xml:lang attribute and its values are described in the XML Recommendation at http://www.w3.org/TR/REC-xml/#sec-lang-tag. Allowable values are language tokens or the null string.

The @translate, @xml:lang, and @dir attributes identify language-specific words or phrases for specific processing (or non-processing, in the case of translate="no").

The cordial response to the question is <q translate="no" xml:lang="de-de" dir="ltr">Nein.</p>

Appendix B.1.2 Architectural attribute group
The architectural attributes group includes a set of attributes defined for document level elements such as <topic> and <map>. These attributes are intended to provide information about the DITA namespace, what level of DITA is in use, and what vocabulary modules are in use.

@DITAArchVersion
Designates the version of the architecture that is in use. The default value will increase with each release of DITA. This attribute is in the namespace "http://dita.oasis-open.org/architecture/2005/". This attribute is defined with the XML data type CDATA, but uses a default value of the current version of DITA. The current default is 2.0.

@domains
Indicates the specialized domains that are included in the grammar file. This attribute is defined with the XML data type CDATA. The value will differ depending on what domains are included in the
current DTD or Schema; a sample value is (topic ui-d) (topic hi-d) (topic pr-d) (topic sw-d) (topic ut-d) (topic indexing-d).

@xmlns:ditaarch
Declares the default DITA namespace. Although this is technically a namespace rather than an attribute, it is included here because it is specified as an attribute in the DTD grammar files distributed by OASIS. The value is fixed to http://dita.oasis-open.org/architecture/2005/.

Appendix B.1.3 Attributes common to many map elements
This attribute group collects several attributes that are used on a variety of map elements. For a few elements, the group is modified slightly to remove an attribute such as @toc or @format; in those cases the element definition will clarify that the element does not use this full set. That is generally done in order to specify a default for one attribute, such as defaulting @format to ditamap on the <mapref> element.

@cascade
Controls how metadata attributes cascade within a map. There are two defined values that should be supported: merge and nomerge.

If no value is set, and no value cascades from an ancestor element, processors SHOULD assume a default of merge.

See Cascading of metadata attributes in a DITA map for more information about how this attribute interacts with metadata attributes.

@chunk
When a set of topics is transformed using a map, the @chunk attribute allows multi-topic documents to be broken into smaller files and multiple individual topics to be combined into larger combined documents.

For a detailed description of the @chunk attribute and its usage, see Chunking.

@collection-type
Collection types describe how links relate to each other. The processing default is unordered, although no default is specified in the DTD or Schema. Allowable values are:

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
Represents a tight grouping in which each of the referenced topics not only relates to the current topic but also relate to each other.

@keyscope
Specifies that the element marks the boundaries of a key scope. See The keyscope attribute (181) for details on how to use the @keyscope attribute.

@linking
Defines some specific linking characteristics of a topic's current location in the 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). Allowable values are:
A topic can only be linked to and cannot link to other topics.

A topic cannot be linked to but can link to other topics.

A topic can be linked to and can link to other topics. Use this to override the linking value of a parent topic.

A topic cannot be linked to or link to other topics.

See Using the -dita-use-conref-target value (177) for more information.

If the @locktitle attribute is set to yes, the content of the <navtitle> element is used for a navigation title, if it is present. If the @locktitle attribute is not present or set to no, the content of the <navtitle> element is ignored, and the title of the referenced topic is used as a navigation title.

Allowable values for @locktitle are:

yes
The content of the <navtitle> element is used for a navigation title.

no
The content of the <navtitle> element is ignored. This is the processing default.

See Using the -dita-use-conref-target value (177) for more information.

Describes the processing role of the referenced topic. The processing default is normal. If no value is specified, but the attribute is specified on an ancestor within a map or within the related-links section, the value will cascade from the closest ancestor. Allowable values are:

normal
Normal topic that is a readable part of the information.

resource-only
The topic is used as a resource for processing purposes. This topic should not be included in a rendered table of contents, and the topic should not be rendered on its own.

See Using the -dita-use-conref-target value (177) for more information.

Describes 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). Allowable values are:

yes

no

See Using the -dita-use-conref-target value (177) for more information.

Specifies whether a topic appears in the table of contents (TOC). 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). Allowable values are:
yes
The topic appears in a generated TOC.

no
The topic does not appear in a generated TOC.

-dita-use-conref-target
See Using the -dita-use-conref-target value (177) for more information.

Appendix B.1.4 Complex-table attribute group

The Complex-table attribute group includes several attributes that are defined on complex table elements. Most of these attributes are part of the OASIS Exchange model; table elements generally use only a subset of the attributes defined in this group. These attributes are not available for the <simpletable> elements.

@align
Describes the alignment of text in a table column. Allowable values are:

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
Use the character specified on the @char attribute for alignment.

-dita-use-conref-target
See Using the -dita-use-conref-target value (177) for more information.

The @align attribute is available on the following table elements: <tgroup>, <colspec>, and <entry>.

@char
Specifies the character for aligning the table entry data.

Default source for <entry> elements starting in this column. If character alignment is specified, the value is the single alignment character source for any implied @char values for entry immediately in this column. A value of "" (the null string) means there is no aligning character.

For example, if align="char" and char="r" are specified, then text in the entry should align with the first occurrence of the letter "r" within the entry.

The @char attribute is available on the following table elements: <colspec> and <entry>.

@charoff
Specifies the horizontal offset of alignment character when align="char".

Default source for <entry> elements starting in this column. For character alignment on an entry in the column, horizontal character offset is the percent of the current column width to the left of the (left edge of the) alignment character.
This value should be number, greater than 0 and less than or equal to 100.
For example, if \texttt{align}="char", \texttt{char}="r", and \texttt{charoff}="50" are all specified, then text in the
entry should align 50\% of the distance to the left of the first occurrence of the character "r" within the
entry.

The \texttt{@charoff} attribute is available on the following table elements: <colspec> and <entry>.

\texttt{@colsep}
Column separator. A value of 0 indicates no separators; 1 indicates separators.

The \texttt{@colsep} attribute is available on the following table elements: <table>, <tgroup>,
<colspec>, and <entry>.

\texttt{@rowheader}
Indicates whether the entries in the respective column \textbf{SHOULD} be considered row headers.
Allowable values are:

\texttt{firstcol}
Indicates that entries in the first column of the table are functionally row headers (analogous to
the way that a \texttt{<thead>} element provides column headers). Applies when \texttt{@rowheader} is used
on the \texttt{<table>} element.

\texttt{headers}
Indicates that entries of a column described using the \texttt{<colspec>} element are functionally row
headers (for cases with more than one column of row headers). Applies when \texttt{@rowheader} is
used on the \texttt{<colspec>} element.

\texttt{norowheader}
Indicates that entries in the first column have no special significance with respect to column
headers. Applies when \texttt{@rowheader} is used on the \texttt{<table>} element.

\texttt{-dita-use-conref-target}
See Using the \texttt{-dita-use-conref-target value} (177) for more information.

\textbf{Note}
This attribute is not part of the OASIS Exchange Table model upon which DITA tables are
based. Some DITA processors or output formats might not support all values.

The \texttt{@rowheader} attribute is available on the following table elements: <table> and <colspec>.

\texttt{@rowsep}
Row separator. A value of 0 indicates no separators; 1 indicates separators.

The \texttt{@rowsep} attribute is available on the following table elements: <table>, <tgroup>, <row>,
<colspec>, and <entry>.

\texttt{@valign}
Indicates the vertical alignment of text in a table entry (cell). Allowable values are:

\texttt{top}
Align the text to the top of the table entry (cell).

\texttt{bottom}
Align the text to the bottom of the table entry (cell).

\texttt{middle}
Align the text to the middle of the table entry (cell).

\texttt{-dita-use-conref-target}
See Using the \texttt{-dita-use-conref-target value} (177) for more information.
The `@valign` attribute is available on the following table elements: `<thead>`, `<tbody>`, `<row>`, and `<entry>.

**Appendix B.1.5 Data-element attributes group**

The data element attributes group includes attributes that are defined for the `<data>` element, and are reused on most or all specializations of the `<data>` element.

- **@datatype**
  Describes 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**
  Defines a unique name for the object.

- **@value**
  Specifies a value associated with the current property or element.

**Appendix B.1.6 Date attributes group**

The date attributes group includes attributes that take date values, and are defined on metadata elements that work with date information.

- **@expiry**
  The date when the information should be retired or refreshed, entered as YYYY-MM-DD, where YYYY is the year, MM is the month from 01 to 12, and DD is the day from 01-31.

- **@golive**
  The publication or general availability (GA) date, entered as YYYY-MM-DD, where YYYY is the year, MM is the month from 01 to 12, and DD is the day from 01-31.

**Appendix B.1.7 Display attribute group**

The display attribute group includes attributes whose values can be used for affecting the display of many elements.

- **@expanse**
  Determines the horizontal placement of the element. Allowable values are:

  - **column**
    Aligns the element with the current column margin

  - **page**
    Places the element on the left page margin for left-to-right presentation, or right page margin for right-to-left presentation.

  - **spread**
    Indicates that, if possible, the object should be rendered across a multi-page spread. If the rendition target does not have anything corresponding to spreads then spread has the same meaning as "page".

  - **textline**
    Aligns the element 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 (177) for more information.

In DITA tables, in place of the `@expanse` attribute used by other DITA elements, the `@pgwide` attribute is used in order to conform to the OASIS Exchange Table Model. The `@pgwide` attribute has a similar semantic (1=page width; 0=resize to galley or column).
Some DITA processors or output formats might not be able to support all values.

@frame
Specifies which portion of a border should surround the element. Allowable values are:

all
  Draw a box around the element
bottom
  Draw a line after the element
none
  Don't draw any lines around this element
sides
  Draw a line at each side of the element
top
  Draw a line before the element
topbot
  Draw a line both before and after the element
@dita-use-conref-target
  See Using the -dita-use-conref-target value (177) for more information.

Some DITA processors or output formats might not be able to support all values.

@scale
Specifies a percentage, selected from an enumerated list, that is used to resize fonts in relation to the normal text size. This attribute is primarily useful for print-oriented display.

The @scale attribute provides an acknowledged style-based property directly on DITA elements. For the <table> and <fig> elements, the intent of the property is to allow authors to adjust font sizes on the content of the containing element, primarily for print accommodation. An <image> in these contexts is to be scaled only by its own direct scale property. If not specifically scaled, such an <image> is unchanged by the scale property of its parent <table> or <fig>.

Allowable values are 50, 60, 70, 80, 90, 100, 110, 120, 140, 160, 180, 200, and -dita-use-conref-target (177). Some DITA processors or output formats might not be able to support all values.

Appendix B.1.8 Link-relationship attribute group
The link relationship attribute group includes attributes whose values can be used for representing navigational relationships. These attributes occur only on elements that represent relationships among DITA elements or between DITA elements and non-DITA resources.

@format
   The @format attribute identifies the format of the resource being referenced. See The format attribute (178) for details on supported values.

@href
   Provides a reference to a resource. See The href attribute (179) for detailed information on supported values and processing implications.

@scope
   The @scope attribute identifies the closeness of the relationship between the current document and the target resource. Allowable values are local, peer, external, and -dita-use-conref-target; see The scope attribute (182) for more information on these values.
@type
Describes the target of a reference. See The type attribute (183) for detailed information on supported values and processing implications.

Appendix B.1.9 Other common attributes
These common attributes are used across a wide variety of elements, with the common definition included below. These attributes are not defined as a group, and many elements only specify one or two from this list.

@anchorref
Identifies a location within another map file where this map will be anchored at runtime. Resolution of the map is deferred until the final step in the delivery of any rendered content. For example, anchorref="map1.ditamap/a1" causes this map to be pulled into the location of the anchor point "a1" inside map1.ditamap when map1.ditamap is rendered for delivery.

@compact
Indicates close vertical spacing between list items. Expanded spacing is the processing default. The output result of compact spacing depends on the processor or browser. Allowable values are:

yes
Indicates compact spacing.

no
Indicates expanded spacing.

@dita-use-conref-target
See Using the -dita-use-conref-target value (177) for more information.

@keyref
@keyref provides a redirectable reference based on a key defined within a map. See The keyref attribute (180) for information on using this attribute.

@mapkeyref
Identifies the map, if any, from which the contained links or metadata are derived. This value might be automatically generated by a process that creates the links or metadata based on map context, as a way to identify which map the material came from. If the <linklist>, or <linkpool>, or metadata is manually created by an author, there is no need to use this attribute. Note that this attribute is not related to the @keyref attribute, and is not used for key based processing.

@xml:space
This attribute is provided on <pre>, <lines>, and on elements specialized from those. It ensures that parsers in editors and transforms respect the white space, including line-end characters, that is part of the data in those elements. It is intended to be part of the default properties of these elements, and not for authors to change or delete. When defined, it has a fixed value of "preserve".

Appendix B.1.10 Simpletable attribute group
The simpletable attribute group includes several attributes that are defined on the <simpletable> element and <simpletable> specializations. These attributes are not defined for the OASIS exchange table (<table>).

@keycol
Defines the column that contains headings for each row. No value indicates no key column. When present, the numerical value causes the specified column to be treated as a vertical header.
@relcolwidth
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%).

Appendix B.1.11 Specialization attributes group
These attributes are designed to be used by specializers, and are not intended for direct use by authors.

@specentry
The specialized entry attribute allows architects of specialized types to define a fixed or default header title for a specialized `<stentry>` element. Not intended for direct use by authors.

@spectitle
The specialized title attribute allows architects of specialized types to define a fixed or default title for a specialized element. Not intended for direct use by authors.

Appendix B.1.12 Topicref-element attributes group
The topicref element attributes group includes attributes that are defined for the `<topicref>` element. These attributes also are used on most or all specializations of the `<topicref>` element.

@copy-to
Use the `@copy-to` attribute on the `<topicref>` element to provide a different resource name for a particular instance of a resource referenced by the `<topicref>` (for example, to separate out the different versions of the topic, rather than combining them on output). If applicable, the `@copy-to` value can include path information. The links and navigation associated with that instance will point to a copy of the topic with the file name you specified.

Applications MAY support `@copy-to` for references to local non-DITA resources.

The `@copy-to` attribute is not supported for references to resources where the effective value for `@scope` is "peer" or "external".

Use the `<linktext>` and `<shortdesc>` in the `<topicref>`'s `<topicmeta>` to provide a unique name and short description for the new copy.

Appendix B.1.13 Complex attribute definitions
Several DITA attributes require more explanation. Those attributes are collected here.

Appendix B.1.13.1 The @conaction attribute
The `@conaction` attribute allows users to push content from one topic into another. It causes the `@conref` attribute to work in reverse, so that the content is pushed from the current topic into another, rather than pulled from another topic into the current one. Allowable values for `@conaction` are: `pushafter`, `pushbefore`, `pushreplace`, `mark`, and `-dita-use-conref-target`.

Note In the descriptions below, the word `target` always refers to the element referenced by a `@conref` attribute.

There are three possible functions using the `@conaction` attribute: replacing an element, pushing content before an element, and pushing content after an element. The `@conaction` attribute always
declares the desired function while the @conref attribute provides the target of the reference using the standard @conref syntax.

In each case, an element pushed using @conref must be of the same type as, or more specialized than, its target. If the pushed element is more specialized than the target, then it should be generalized when the @conref is resolved. This ensures that the content will be valid in the target topic.

- It is valid to push using @conref when the two elements involved are of the same type. For example, a <step> element can use the conref push feature with another <step> as the target of the @conref.
- The target element can be more general than the source. For example, it is legal to push a <step> element to replace a general list item (<li>); the <step> element should be generalized back to a list item during the process.
- It is not possible to push a more general element into a specialized context. For example, it is not legal to push a list item (<li>) in order to replace a <step>, because the list item allows many items that are not valid in the specialized context.

**Replacing content in another topic**

When the @conaction attribute is set to "pushreplace", the source element will replace the target specified on the @conref attribute. The pushed content remains in the source topic where it was originally authored.

For example, assume that a task in example.dita has the id "example", and contains a <step> with the id "b":

```
<task id="example" xml:lang="en"
<title>Example topic</title>
<taskbody>
  <steps>
    <step id="a"><cmd>A</cmd></step>
    <step id="b"><cmd>B</cmd></step>
    <step id="c"><cmd>C</cmd></step>
  </steps>
</taskbody>
</task>
```

In order to replace the step with id="b", another topic must combine a @conaction value of "pushreplace" with a @conref attribute that references this <step>:

```
<task id="other" xml:lang="en">
  ...
  <step conaction="pushreplace"
    conref="example.dita#example/b">
    <cmd>Updated B</cmd>
  </step>
  ...
</task>
```

The result will be an updated version of example.dita which contains the pushed <step>:

```
<task id="example" xml:lang="en"
<title>Example topic</title>
<taskbody>
  <steps>
    <step id="a"><cmd>A</cmd></step>
    <step id="b"><cmd>Updated B</cmd></step>
    <step id="c"><cmd>C</cmd></step>
  </steps>
</taskbody>
</task>
```
When resolving a conref push action, attributes are resolved using the same precedence as for normal @conref, with one exception. Attributes on the element with the @conref attribute (in this case, the source doing the push) will take priority over those on the referenced element. The exception is that if the source element does not specify an ID, the ID on the referenced element remains; if the source element does specify an ID then that replaces the ID on the referenced element.

It is an error for two source topics to replace the same element. Applications MAY warn users if more than one element attempts to replace a single target.

### Pushing content before or after another element

Setting the @conaction attribute to "pushbefore" allows an element to be pushed before the element referenced by the @conref attribute. Likewise, setting the @conaction attribute to "pushafter" allows an element to be pushed after the element referenced by the @conref attribute. Multiple sources can push content before or after the same target; the order in which that content is pushed is undefined.

When an element is pushed before or after a target, the resulting document will have at least two of that element. Because this is not always valid, a document attempting to push content before or after a target must take an extra step to ensure that the result will be valid. The extra step makes use of the conaction="mark" value.

When pushing before, the @conref attribute itself looks just as it did when replacing, but the @conaction attribute is set to "mark" because it is marking the target element. This element remains empty; its purpose is to ensure that it is legal to have more than one of the current element. Immediately before the element which marks the target, you will place the content that you actually want to push. This element will set the @conaction attribute to "pushbefore".

When pushing after, the procedure is the same, except that the order of the elements is reversed. The element with conaction="pushafter" comes immediately after the element which marks the target.

Attributes on the element which is pushed (the one with conaction="pushbefore") must be retained on the target, apart from the @conaction attribute itself. If this causes the result document to end up with duplicate IDs, an application can recover by dropping the duplicate ID, modifying it to ensure uniqueness, or warning the user.

The following restrictions apply when pushing content before or after an element:

- The elements that use conaction="mark" and conaction="pushbefore" are the same type as each other and appear in sequence. This restriction prevents a topic from trying to push a <body> element before or after another <body> element, because it is not valid to have two body elements in sequence.
- Either the container elements of the source and target match, or the container of the source element is a specialization of the target's container. This is also to ensure validity of the target; for example, while it is possible to include multiple titles in a <section>, it is not possible to do so in a figure. Comparing the parents prevents a second <section> title from being pushed before a figure title (the resulting figure would not be valid DITA). This restriction only applies to the pushbefore or pushafter actions, not to the pushreplace action.

When content is pushed from one topic to another, it is still rendered in the original context. Processors might delete the empty element that has the conaction="mark" attribute. In order to push content from a topic without actually rendering that topic on its own, the topic should be referenced from the map with the @processing-role attribute set to "resource-only".
Example: pushing an element before the target

The following example pushes a `<step>` before "b" in the `example.dita` file shown above.

```xml
<step conaction="pushbefore"><cmd>Do this before B</cmd></step>
<step conaction="mark" conref="example.dita#example/b"></cmd></step>
```

The result contains the pushed `<step>` element before "b".

```xml
<task id="example" xml:lang="en">
  <title>Example topic</title>
  <taskbody>
    <steps>
      <step id="a"><cmd>A</cmd></step>
      <step><cmd>Do this before B</cmd></step>
      <step id="b"><cmd>B</cmd></step>
      <step id="c"><cmd>C</cmd></step>
    </steps>
  </taskbody>
</task>
```

Example: pushing an element after the target

Pushing an element after a target is exactly the same as pushing before, except that the order of the "mark" element and the pushed element are reversed.

```xml
<step conaction="mark" conref="example.dita#example/b"></cmd></step>
<step conaction="pushafter"><cmd>Do this AFTER B</cmd></step>
```

In this case the resulting document has the pushed content after `<step>` b:

```xml
<task id="example" xml:lang="en">
  <title>Example topic</title>
  <taskbody>
    <steps>
      <step id="a"><cmd>A</cmd></step>
      <step id="b"><cmd>B</cmd></step>
      <step><cmd>Do this AFTER B</cmd></step>
      <step id="c"><cmd>C</cmd></step>
    </steps>
  </taskbody>
</task>
```

Combining `@conaction` with `@conkeyref` or `@conrefend`

The `@conkeyref` attribute can be used as an indirect way to specify a `@conref` target. If the `@conkeyref` attribute is specified on an element that also uses the `@conaction` attribute, the `@conkeyref` attribute is used to determine the target of the conref push (as it would normally be used to determine the target of `@conref`).

The conref push function does not provide the ability to push a range of elements, so it is an error to specify the `@conrefend` attribute together with the `@conaction` attribute. If the two are specified together an application can recover by warning the user, ignoring the `@conrefend` attribute, or with some other implementation strategy.
Appendix B.1.13.2 The `@conrefend` attribute
The `@conrefend` attribute is used when referencing a range of elements with the conref mechanism. The `@conref` or `@conkeyref` attribute references the first element in the range, while `@conrefend` references the last element in the range.

Using `@conref` together with `@conrefend`

The following markup rules apply when using or implementing `@conrefend`:

- The start and end elements of a range **MUST** be of the same type as the referencing element or generalizable to the referencing element.
- The start and end elements in a range **MUST** share the same parent, and the start element **MUST** precede the end element in document order.
- The parent of the referencing element **MUST** be the same as the parent of the referenced range or generalizable to the parent of the referencing element.

In addition, several other items must be taken into account:

- Processors will resolve the range by pulling in the start target and following sibling XML nodes across to and including the end target.
- As with `@conref`, if the `@conrefend` references a more specialized version of the referencing element, applications should generalize the target when resolving.
- It is not valid to use `@conrefend` to reference a more general version of an element (such as using `<step>` to reference an `<li>` element).
- Other nodes (such as elements or text) between the start and end of a range do not have to match the referencing element.
- With single conref, an `@id` attribute from the referenced element will not be preserved on the resolved content. With a range, an `@id` on both the start and the end elements will not be preserved. `@id` attributes on intermediate or child nodes should be preserved; if this results in duplicate `@id` values, an application can recover by changing the `@id`, warning the user, or implementing another strategy.
- With a single conref, attributes specified on the referencing element can be used to override attributes on the referenced element. With a conref range, the same is true, with the following clarifications:
  - When an `@id` attribute is specified on the referencing element, it will only be preserved on the first element of the resolved range.
  - When other attributes are specified, they will only apply to referenced elements of the same type. For example, if `<step>` is used to pull in a range of sequential `<step>` elements, locally specified attributes apply to all steps in the range. If `<ol>` is used to pull in a series of `<ol>, <p>, <ol>`, locally specified attributes apply only to the `<ol>` elements in that range.

Example: reusing a set of list items

**Figure 9: List example: Source topic.dita with ids**

```xml
<topic id="x">
  ...
  <body>
    <ol>
      <li id="apple">A</li>
      <li id="bear">B</li>
      <li id="cat">C</li>
      <li id="dog">D</li>
    </ol>
  </body>
</topic>
```
Figure 10: List example: Reusing topic with conrefs

```xml
<topic id="y">
  ...
  <body>
    <ol>
      <li>My own first item</li>
      <li conref="topic.dita#x/bear" conrefend="topic.dita#x/dog"/>
      <li>And a different final item</li>
    </ol>
  </body>
</topic>
```

Figure 11: List example: Processed result of reusing topic

```xml
<topic id="y">
  ...
  <body>
    <ol>
      <li>My own first item</li>
      <li>B</li>
      <li id="cat">C</li>
      <li>D</li>
      <li>And a different final item</li>
    </ol>
  </body>
</topic>
```

Example: Reusing a set of blocks

**Figure 12: Block level example: Source topic.dita with ids**

```xml
<topic id="x">
  ...
  <body>
    <p id="p1">First para</p>
    <ol id="mylist">
      <li id="apple">A</li>
      <li id="bear">B</li>
      <li id="cat">C</li>
      <li id="dog">D</li>
      <li id="eel">E</li>
    </ol>
    <p id="p2">Second para</p>
  </body>
</topic>
```

**Figure 13: Block level example: Reusing topic with conrefs**

```xml
<topic id="y">
  ...
  <body>
    <p conref="topic.dita#x/p1" conrefend="topic.dita#x/p2"/>
  </body>
</topic>
```
Using @conrefend together with @conkeyref

When the @conkeyref attribute is used in place of @conref, a key is used to address the target of the reference. The @conrefend attribute, which indicates the end of a @conref range, cannot use a key. Instead the the map or topic element addressed by the key name component of the @conkeyref is used in place of whatever map or topic element is addressed by the @conrefend attribute.

For example, if the value of the @conkeyref attribute is "config/step1" and the value of the @conrefend is defaultconfig.dita#config/laststep, the conref range will end with the step that has id="laststep" in whatever topic is addressed by the key name "config". If the key name "config" is not defined, and the @conref attribute itself is not present for fallback, the @conrefend attribute is ignored.

Example: Combining @conrefend with @conkeyref

In this example the key "xmp" is defined as the first topic in the file examples.dita.

To reuse these list items by using the key, the @conkeyref attribute combines the key itself with the sub-topic id (first) to define the start of the range. The @conrefend attribute defines a default high-level object along with the sub-topic id (last) that ends the range:
The @conkeyref attribute uses a key to reference the first topic in examples.dita, so the range begins with the object examples.dita#examples/first. The high-level object in the @conrefend attribute (default.dita#default) is replaced with the object represented by the key (the first topic in examples.dita), resulting in a range that ends with the object examples.dita#examples/last.

**Figure 16: Combining @conref, @conkeyref, and @conrefend**

When @conref, @conkeyref, and @conrefend are all specified, the key value takes priority.

```
<li conkeyref="thisconfig/start"
    conref="standardconfig.dita#config/start"
    conrefend="standardconfig.dita#config/end"/>
```

- If the key "thisconfig" is defined as mySpecialConfig.dita#myconfig, then the range will go from the list item with id="start" to the list item with id="end" in the topic mySpecialConfig.dita#myconfig.
- If the key "thisconfig" is defined as myConfig.dita, then the range will go from the list item with id="start" to the list item with id="end" within the first topic in myConfig.dita.
- If the key "thisconfig" is not defined, then the unchanged @conref and @conrefend attributes are used as fallback. In that case, the range will go from the list item with id="start" to the list item with id="end" within the topic standardconfig.dita#config.

**Error conditions**

When encountering an error condition, an implementation can issue an error message.

<table>
<thead>
<tr>
<th>Condition or Issue</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>The @conref attribute cannot be resolved in the target document (the target element might have been removed or its id has changed).</td>
<td>The @conref is ignored.</td>
</tr>
<tr>
<td>The @conrefend attribute cannot be resolved in the target document (the target element might have been removed or its id has changed).</td>
<td>Range cannot be resolved, optional recovery processes the result as a simple conref.</td>
</tr>
<tr>
<td>Start and end elements are not siblings in the target document.</td>
<td>If the start element exists, optional recovery processes the result as a simple conref.</td>
</tr>
<tr>
<td>End element occurs before the start element in the target document.</td>
<td>If the start element exists, optional recovery processes the result as a simple conref.</td>
</tr>
<tr>
<td>An element has a @conrefend attribute but is missing the @conref attribute.</td>
<td>No result.</td>
</tr>
</tbody>
</table>

**Appendix B.1.13.3 The @conkeyref attribute**

The @conkeyref attribute provides an indirect content reference to topic elements, map elements, or elements within maps or topics. When the DITA content is processed, the key references are resolved using key definitions from DITA maps.

For content references from map elements to map elements or topic elements to topic elements, the value of the @conkeyref attribute is a key name, where the key must be bound to a map element (for references from map elements) or a topic element (for references from topic elements). For all other elements, the value of the @conkeyref attribute is a key name, an optional slash ("/"), and the ID of the target element, where the key name must be bound to the map or topic that contains the topic element.
When the key name specified by the @conkeyref attribute is not defined and the element also specifies a @conref attribute, the @conref attribute is used to determine the content reference relationship. If no @conref attribute is specified there is no content reference relationship.

Processors SHOULD issue a warning when a @conkeyref reference cannot be resolved and there is no @conref attribute to use as a fallback. Processors MAY issue a warning when a @conkeyref cannot be resolved to an element and a specified @conref is used as a fallback.

The @conrefend attribute, which defines the end of a conref range, cannot include a key. Instead the map or topic element addressed by the key name component of the @conkeyref is used in place of whatever map or topic element is addressed by the @conrefend attribute. See Using conrefend together with conkeyref (175) for more information and for examples of this behavior.

Appendix B.1.13.4 The @conref attribute

The @conref attribute is used to reference content that can be reused. It allows reuse of DITA elements, including topic or map level elements.

The value of the @conref attribute must be a URI reference to a DITA element. See URI-based (direct) addressing for details on specifying URI references to DITA elements. As with other DITA references, a @conref attribute that references a resource without an ID is treated as a reference to the first topic or map in the document.

Note When using the @conref attribute on an element, the content of that element is ignored. For example, if a phrase is marked up like this:

```xml
<ph conref="#topic/ph">Something</ph>
```

the word “Something” will be replaced by the content of the referenced <ph> element.

Appendix B.1.13.4.1 Using the -dita-use-conref-target value

The value -dita-use-conref-target is available on enumerated attributes and can also be specified on other attributes. When an element uses @conref to pull in content, for any of its attributes assigned a value of -dita-use-conref-target, the resulting value for those attributes is also pulled in from the referenced element.

Ordinarily, when an element uses @conref, any other attributes specified locally will be preserved when the reference is resolved. This causes problems when attributes are required, because required attributes must be specified regardless of whether the @conref attribute is present. The purpose of the -dita-use-conref-target value is to allow the author to specify a value for a required attribute while still allowing the conref resolution process to use the matching attribute from the referenced element. The value has the same result when the attribute is not required.

The -dita-use-conref-target token is allowed on any attribute where it is not prohibited by the XML grammar files or by the specification. For example, while @cols on the <tgroup> element is defined as being a number, this token is implicitly allowed in order to support conref processing for <tgroup>. However, the token is not allowed for the @id attribute on the <topic> element, because "-dita-use-conref-target" does not fit the syntax required by the XML grammar files.

This example shows a DITA 1.3 map where the <topichead> element uses @conref. It specifies the @navtitle attribute as well as the @toc attribute. In the resolved element, @navtitle from the referencing element is not preserved because it uses -dita-use-conref-target. The @toc attribute from the referencing element overrides the @toc attribute on the referenced element using normal conref resolution rules.
Note In earlier versions of DITA, @navtitle was required on the <topichead> element. While it is no longer required, the example still illustrates the expected DITA 1.3 processing for both required and non-required attributes.

Figure 17: Pre-resolution

```xml
<map>
<title>Conref demonstration</title>
<topichead id="heading"
    navtitle="This is a heading"
    toc="yes"
    linking="normal">
    <topicref href="topic.dita" navtitle="content"/>
</topichead>
<topichead conref="#heading"
    navtitle="-dita-use-conref-target"
    toc="no">
</topichead>
</map>
```

Figure 18: Effective result post-resolution

```xml
<map>
<title>Conref demonstration</title>
<topichead id="heading"
    navtitle="This is a heading"
    toc="yes"
    linking="normal">
    <topicref href="topic.dita" navtitle="content"/>
</topichead>
<topichead navtitle="This is a heading"
    toc="no"
    linking="normal">
    <topicref href="topic.dita" navtitle="content"/>
</topichead>
</map>
```

Appendix B.1.13.5 The @format attribute

The @format attribute identifies the format of the resource that is referenced. If no value is specified, but the attribute is specified on an ancestor within a map or within the related-links section, the value will cascade from the closest ancestor.

The following values for @format have special processing implications:

- **dita**
  The destination uses DITA topic markup or markup specialized from a DITA topic. Unless otherwise specified, when @format is set to "dita", the value for the @type attribute will be treated as "topic".

- **ditamap**
  The linked-to resource is a DITA map. It represents the referenced hierarchy at the current point in the referencing map. References to other maps can occur at any point in a map, but because relationship tables are only valid as children of a map, referenced relationship tables are treated as children of the referencing map.

  **Note** If a <topicref> element that references a map contains child <topicref> elements, the processing behavior regarding the child <topicref> elements is undefined.

- **(no value)**
  The processing default is used. The processing default for the @format attribute is determined by inspecting the value of the @href attribute. If the @href attribute specifies a file extension, the processing default for the @format attribute is that extension, after conversion to lower-case and with no leading period. The only exception to this is if the extension is "xml", in which case the default
If the actual format of the referenced content differs from the effective value of the @format attribute, and a processor is capable of identifying such cases, it MAY recover gracefully and treat the content as its actual format, but SHOULD also issue a message.

For other formats, using the file extension without the "." character typically represents the format. For example, the following values are all possible values for @format:

- **html**: The format of the linked-to resource is HTML or XHTML.
- **pdf**: The format of the linked-to resource is PDF.
- **txt**: The format of the linked-to resource is a text file.

### Appendix B.13.6 The @href attribute

The @href attribute is used to reference another DITA topic or map, a specific element inside a DITA topic or map, an external Web page, or another non-DITA resource.

The value of a DITA @href attribute must be a valid URI reference [RFC 3986].

If the value of the @href attribute is not a valid URI reference, an implementation MAY generate an error message; it MAY recover from this error condition by attempting to convert the value to a valid URI reference.

Note that the path separator character in a URI is the forward slash ("/"); the backward slash character ("\") is not permitted unescaped within URIs.

When an @href attribute references a DITA resource, an @href value that consists of a URI without a fragment identifier resolves to the document element in the referenced document. For the purposes of rendering, such as when a <topicref> reference to a DITA document is used to render the content as HTML, this means that all topics (and topic specializations) in the target document are included in the reference. For the purpose of linking, the reference resolves to the first (or only) topic (or topic specialization) in the document.

An @href value that consists of a URI with a fragment identifier must have a DITA local identifier as the portion after the hash. A DITA local identifier consists of topicID/elementID for a subelement of a topic, and of elementID for topics, maps, and subelements of a map. If the topic referenced by a DITA local identifier is for the same topic, then topicID can be replaced by a period; see Processing xrefs and conrefs within a conref for more information on how this syntax relates to conref resolution.

Note that certain characters - including but not limited to the hash sign ("#"), question mark ("?"), back slash ("\"), and space - are not permitted unescaped within URIs. Such characters must be percent-encoded. Also note that the ampersand ("&") and less than ("<") characters are not permitted in XML attribute values; they must be represented by appropriate character or entity references. Some tools might perform this encoding automatically, while other tools might require that users either avoid the special characters or manually insert the encoding.

### Example: Common syntax for the @href attribute

The following table includes some examples of common @href syntax. Note that these examples represent only a few common scenarios and are not all inclusive.
<table>
<thead>
<tr>
<th>Target</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>The first topic in a DITA document</td>
<td>href=&quot;file.dita&quot;</td>
</tr>
<tr>
<td>A specific topic in a DITA document</td>
<td>href=&quot;file.dita#topicid&quot;</td>
</tr>
<tr>
<td>A non-topic element inside a DITA topic</td>
<td>href=&quot;#topicid/elementid&quot;</td>
</tr>
<tr>
<td>A non-topic element inside the same DITA topic as the reference</td>
<td>href=&quot;#./elementid&quot;</td>
</tr>
<tr>
<td>An element in a DITA map</td>
<td>href=&quot;myMap.ditamap#map-branch&quot;</td>
</tr>
<tr>
<td>An image</td>
<td>href=&quot;exampleImage.jpg&quot;</td>
</tr>
<tr>
<td>An external resource</td>
<td>href=&quot;<a href="http://www.example.org">http://www.example.org</a>&quot;</td>
</tr>
</tbody>
</table>

where:

- `topicid` is the value of the @id attribute on the DITA topic.
- `elementid` is the value of the @id attribute on the (non-topic) DITA element.
- `map-branch` is the value of the @id attribute on the DITA map element.

**Appendix B.1.13.7 The @keyref attribute**

The @keyref attribute provides an indirect, late-bound reference to topics, to collections of topics (ditabase), to maps, to referenceable portions of maps, to non-DITA documents, to external URIs, or to XML content contained within a key definition topic reference. When the DITA content is processed, the key references are resolved using key definitions from DITA maps.

For elements that only refer to topics or non-DITA resources, the value of the @keyref attribute is a key name. For elements that can refer to elements within maps or topics, the value of the @keyref attribute is a key name, a slash ("/"), and the ID of the target element, where the key name must be bound to either the map or topic that contains the target element.

**Appendix B.1.13.8 The @keys attribute**

A @keys attribute consists of one or more space-separated keys. Map authors define keys using a <topicref> or <topicref> specialization that contains the @keys attribute. Each key definition introduces an identifier for a resource referenced from a map. Keys resolve to the resources given as the @href value on the key definition <topicref> element, to content contained within the key definition <topicref> element, or both.

The @keys attribute uses the following syntax:

- The value of the @keys attribute is one or more space-separated key names.
- Key names consist of characters that are legal in a URI. The case of key names is significant.
- The following characters are prohibited in key names: "{", "}"", "[", "]", "/", ",", ",", ",", and whitespace characters.

A key cannot resolve to sub-topic elements, although a @keyref attribute can do so by combining a key with a sub-topic element id.
Appendix B.1.13.9 The @keyscope attribute

The @keyscope attribute consists of one or more space-separated key scope names. Map authors define the boundaries for key scopes by specifying the @keyscope attribute on <map> elements, <topicref> elements, or elements that are specializations of <map> or <topicref>. Such elements, their contents, and any locally-scoped content referenced from within the element, are considered to be part of the scope. Keys defined within a scope are only directly referenceable from within the same scope. They can be referenced from the parent scope using the scope's name, followed by a period, followed by the key name.

All key scopes are contiguous and non-intersecting. Within a root map, two distinct key scopes with the same name have no relationship with each other aside from that implied by their relative locations in the key scope hierarchy. They do not, for example, share key definitions. The only processing impact of a key scope's names is in defining the prefixes used when contributing qualified key names to the parent scope. For example, consider the following map segment:

```
<map>
  <topicgroup keyscope="xyz" id="scope1">
    <keydef keys="a" id="def1"/>
    <!-- other topic references -->
  </topicgroup>
  <topicgroup keyscope="xyz" id="scope2">
    <keydef keys="a" id="def2"/>
    <!-- other topic references -->
  </topicgroup>
  <!-- lots of other content -->
</map>
```

This map creates two distinct scopes that happen to use the same name (xyz). This results in the following:

- Each <topicgroup> sets a scope of "xyz" and includes a key "a". From outside of those two scopes, references to keyref="xyz.a" (key "a" within the scope "xyz") will always resolve to the first instance of that value, which is in the first <topicgroup>.
- Within the first <topicgroup>, content uses keyref="a" will resolve to the key in that branch (defined on the element with id="def1").
- Within the second <topicgroup>, content uses keyref="a" will resolve to the key in that branch (defined on the element with id="def2").

Appendix B.1.13.10 The @role and @otherrole attributes

The @role attribute defines the role the target 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 structure could be used to sort and classify links at display time.

Supported values for @role

Allowable values for the @role attribute are:

- **parent**: Indicates a link to a topic that is a parent of the current topic.
- **child**: Indicates a link to a direct child such as a directly nested or dependent topic.
- **sibling**: Indicates a link between two children of the same parent 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.

previous
Indicates a link to the previous topic in a sequence.

cousin
Indicates a link to another topic in the same hierarchy that is not a parent, child, sibling, next, or previous.

ancestor
Indicates a link to a topic above the parent topic.

descendant
Indicates a link to a topic below a child topic.

other
Indicates any other kind of relationship or role. Enter that role as the value for the @otherrole attribute.

-dita-use-conref-target
See Using the -dita-use-conref-target value (177) for more information.

The @otherrole attribute is available to specify an alternate role that is not available in the list above, and should be used in conjunction with role="other".

Appendix B.1.13.11 The @scope attribute
The @scope attribute identifies the closeness of the relationship between the current document and the target resource.

- Set @scope to "local" when the resource is part of the current set of content.
- Set @scope to "peer" when the resource is part of the current set of content but might not be accessible at build time, or for maps to be treated as root maps for the purpose of creating map-to-map key references (peer maps). An implementation might open such resources in the same browser window to distinguish them from those with @scope set to "external".
- Set @scope to "external" when the resource is not part of the current information set and should open in a new browser window.
- See Using the -dita-use-conref-target value (177) for more information on -dita-use-conref-target.

If no value is specified, but the attribute is specified on an ancestor within a map or within the related-links section, the value will cascade from the closest ancestor. The processing default is determined by the value of the @href attribute. In most cases, the processing default is "local". However the processing default is "external" whenever the absolute URI in the @href attribute begins with one of the following schemes:

- http
- https
- ftp
- mailto

Processors can consider additional URI schemes as "external" by default. Processors MUST always consider relative URIs as "local" by default.
Appendix B.13.12 The @type attribute

The @type attribute is used on linking elements to describe the target of a cross-reference. It also is used on the <note> element to describe the note type, as well as on several other elements for varying purposes.

The descriptions for the @type attribute on linking elements and on <note> are included in this section; for other elements, such as <audience>, <copyright>, and <object>, the description can be found with the topic for the specific element.

Using @type on a linking element

The @type attribute describes the target of a cross-reference and might generate cross-reference text based on that description. Only the <xref> element can link to content below the topic level: other types of linking can target whole topics, but not parts of topics. Typically <xref> should also be limited to topic-level targets, unless the output is primarily print-oriented. Web-based referencing works best at the level of whole topics, rather than anchor locations within topics.

If not explicitly specified on an element, the @type attribute value cascades from the closest ancestor element. If there is no explicit value for the @type attribute on any ancestor, a default value of “topic” is used.

During output processing for references to DITA topics (format="dita"), it is an error if the actual type of a DITA topic and the explicit, inherited, or default value for the @type attribute are not the same as or a specialization of the @type attribute value. In this case, an implementation MAY give an error message, and MAY recover from this error condition by using the @type attribute value.

During output processing for references to non-DITA objects (that is, either scope is “external” or format is neither “dita” nor “ditamap”) or other cases where the type of the referenced item cannot be determined from the item itself, the explicit, inherited, or default value for the @type attribute is used without any validation.

When a referencing element is first added to or updated in a document, DITA-aware editors MAY set the @type attribute value based on the actual type of a referenced DITA topic. If the @type attribute is specified when referencing DITA content, it should match one of the values in the referenced element's @class attribute. The @type value can be an unqualified local name (for example, "fig") or a qualified name exactly as specified in the @class attribute (for example, "mymodule/mytype"). Processors might ignore qualified names or consider only the local name.

For example, if the value is set to type="topic", the link could be to a generic topic, or any specialization of topic, including concept, task, and reference. Applications MAY issue a warning when the specified or inherited @type attribute value does not match the target (or a specialization ancestor of the target).

Some possible values for use on the <xref> element and its specializations include:

- **fig**
  Indicates a link to a figure.

- **table**
  Indicates a link to a table.

- **li**
  Indicates a link to an ordered list item.

- **fn**
  Indicates a link to a footnote.
section
   Indicates a link to a section.

Other values that can be used on any linking element include:

concept, task, reference, topic
   Cross-reference to a topic type.

(no value)
   The processor should retrieve the actual type from the target if available. If the type cannot be
determined, the default should be treated as "topic".

-dita-use-conref-target
   See Using the -dita-use-conref-target value (177) for more information.

Other values can be used to indicate other types of topics or elements as targets. Processing is only
required to support the above list or specializations of types in that list. Supporting additional types as
targets might require the creation of processing overrides.

Using @type in a <note> element

In a <note> element, this defines the type of note. For example, if the note is a tip, the word Tip might be
used to draw the reader's attention to it. The values danger, warning, and notice have meanings that are
based on ANSI Z535 and ISO 3864 regulations.

If @type is set to "other", the value of the @othertype attribute can be used. If you use @othertype,
many processors will require additional information on how to process the value. Allowable values for the
@type attribute are:

note
   This is just a note.

attention
   Please pay extra attention to this note.

caution
   Care is required when proceeding.

danger
   Important! Be aware of this before doing anything else. When used with the <hazardstatement>
element, this indicates an imminently hazardous situation which, if not avoided, will result in death or
serious injury.

fastpath
   This note will speed you on your way.

important
   This note is important.

notice
   Indicates a potential situation which, if not avoided, might result in an undesirable result or state.

remember
   Don't forget to do what this note says.

restriction
   You can't do what this note says.

tip
   This is a fine little tip.
warning
Indicates a potentially hazardous situation. When used with the `<hazardstatement>` element, this indicates a situation which, if not avoided, could result in death or serious injury.

trouble
Provides information about how to remedy a trouble situation.
other
This is something other than a normal note.

-dita-use-conref-target
See Using the -dita-use-conref-target value (177) for more information.

Appendix B.2 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.

Table 2: Formatting expectations for DITA elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Suggested formatting</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;chdeschd&gt;</code></td>
<td>Apply bold highlighting to the contents of the <code>&lt;chdeschd&gt;</code> element.</td>
</tr>
<tr>
<td><code>&lt;choicetable&gt;</code></td>
<td>If choice table does not contain a header row, processors typically auto-generate labels for the columns in the choice table. Unless the <code>@keycol</code> attribute is set to 0, processors typically apply bold highlighting to the contents of the &quot;Option&quot; column.</td>
</tr>
<tr>
<td><code>&lt;choptionhd&gt;</code></td>
<td>Apply bold highlighting to the contents of the <code>&lt;choptionhd&gt;</code> element.</td>
</tr>
<tr>
<td><code>&lt;codeblock&gt;</code></td>
<td>Use a monospaced font for the contents of the <code>&lt;codeblock&gt;</code> element.</td>
</tr>
<tr>
<td><code>&lt;codeph&gt;</code></td>
<td>Use a monospaced font for the contents of the <code>&lt;codeph&gt;</code> element.</td>
</tr>
<tr>
<td><code>&lt;menucascade&gt;</code></td>
<td>Separate <code>&lt;uicontrol&gt;</code> elements with a character to represent the menu cascade.</td>
</tr>
<tr>
<td><code>&lt;numcharref&gt;</code></td>
<td>Surround the contents of the <code>&lt;numcharref&gt;</code> element with a leading ampersand (&amp;) and a trailing semi-colon (;).</td>
</tr>
<tr>
<td><code>&lt;parameterentity&gt;</code></td>
<td>Surround the contents of the <code>&lt;numcharref&gt;</code> element with a leading percentage sign (%) and a trailing semi-colon (;).</td>
</tr>
<tr>
<td><code>&lt;screen&gt;</code></td>
<td>Enclose the contents of the <code>&lt;screen&gt;</code> element with a box to suggest a computer display screen.</td>
</tr>
<tr>
<td><code>&lt;shortcut&gt;</code></td>
<td>Highlight the keyboard shortcut with underlining.</td>
</tr>
<tr>
<td><code>&lt;steps&gt;</code></td>
<td>Render steps with only a single step as a paragraph. Render two or more steps as an ordered list.</td>
</tr>
<tr>
<td><code>&lt;syntaxdiagram&gt;</code></td>
<td>Traditionally, the syntax diagram is formatted with &quot;railroad tracks&quot; that connect the units of the syntax together, but the presentation might differ depending on the output media.</td>
</tr>
<tr>
<td><code>&lt;textentity&gt;</code></td>
<td>Surround the contents of the <code>&lt;textentity&gt;</code> element with a leading ampersand (&amp;) and a trailing semi-colon (;).</td>
</tr>
<tr>
<td><code>&lt;var&gt;</code></td>
<td>Apply italic highlighting to the contents of the <code>&lt;var&gt;</code> element.</td>
</tr>
</tbody>
</table>

This topic contains a list of all OASIS DITA elements that are available in the edition. 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.

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.

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.
- 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.
- 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.
- 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. 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.
- If an element has translatable attributes, they are listed in the last column. Note that the @spectitle and @specentry attributes are described with a footnote.

<table>
<thead>
<tr>
<th>Element</th>
<th>Suggested formatting</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;xmlatt&gt;</code></td>
<td>Precede the contents of the <code>&lt;xmlatt&gt;</code> element with a commercial at symbol (@) or equals character (=).</td>
</tr>
<tr>
<td><code>&lt;xmelement&gt;</code></td>
<td>Surround the contents of the <code>&lt;xmelement&gt;</code> element with leading (&lt;) and trailing (&gt;) angle brackets.</td>
</tr>
<tr>
<td><code>&lt;&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>

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

2. The @spectitle and @specentry attributes can contain translatable text. The direct use of fixed-in-the-DTD text by tools is discouraged, in favor of using the value as a lookup string to find the translation outside of the file, using accepted localization methods for generated text.
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

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;abstract&gt;</code></td>
<td>N/A</td>
<td>block**</td>
<td>block**</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;alt&gt;</code> <strong>Footnote</strong></td>
<td>N/A</td>
<td>block**</td>
<td>block**</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;author&gt;</code></td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;body&gt;</code></td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;bodydiv&gt;</code> <strong>(new in DITA 1.2)</strong></td>
<td>N/A</td>
<td>block**</td>
<td>block**</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;boolean&gt;</code></td>
<td>N/A</td>
<td>inline**</td>
<td>inline**</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td><code>&lt;brand&gt;</code></td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;category&gt;</code></td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;cite&gt;</code></td>
<td>N/A</td>
<td>inline**</td>
<td>inline**</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;colspec&gt;</code></td>
<td>N/A</td>
<td>n/a**</td>
<td>n/a**</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td><code>&lt;component&gt;</code></td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;copyrholder&gt;</code></td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;copyright&gt;</code></td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;copyryear&gt;</code></td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;created&gt;</code></td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;critdates&gt;</code></td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;data&gt;</code></td>
<td>N/A</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no (likely to change for some specializations)</td>
<td></td>
</tr>
<tr>
<td><code>&lt;data-about&gt;</code></td>
<td>N/A</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;dd&gt;</code></td>
<td>N/A</td>
<td>block**</td>
<td>block**</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;ddhd&gt;</code></td>
<td>N/A</td>
<td>block**</td>
<td>block**</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

---

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

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 use of the `@alt` attribute is deprecated in favor of the `<alt>` element.
<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;desc&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;div&gt; (new in DITA 1.3)</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;dl&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitleFootnote.</td>
</tr>
<tr>
<td>&lt;dlentry&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;dhead&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;draft-comment&gt;</td>
<td>N/A</td>
<td>block***Footnote.</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;dt&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;dthd&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;entry&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;example&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitleFootnote.</td>
</tr>
<tr>
<td>&lt;featnum&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;fig&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitleFootnote.</td>
</tr>
<tr>
<td>&lt;figgroup&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;fn&gt;</td>
<td>N/A</td>
<td>block***Footnote.</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;foreign&gt;</td>
<td>N/A</td>
<td>blockFootnote.</td>
<td>blockFootnote.</td>
<td>noFootnote.</td>
<td></td>
</tr>
<tr>
<td>&lt;image&gt;</td>
<td>N/A</td>
<td>block when @placement=break, otherwise inline</td>
<td>block when @placement=break, otherwise inline</td>
<td>yes</td>
<td>@altFootnote.</td>
</tr>
<tr>
<td>&lt;index-base&gt;</td>
<td>N/A</td>
<td>block***Footnote.</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;index-sort-as&gt; (new in DITA 1.2)</td>
<td>N/A</td>
<td>block***Footnote.</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;indexterm&gt;</td>
<td>N/A</td>
<td>block***Footnote.</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;indextermref&gt;</td>
<td>N/A</td>
<td>inline</td>
<td>n/a</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>&lt;itemgroup&gt;</td>
<td>N/A</td>
<td>inline</td>
<td>inline (except when within &lt;keywords&gt; – see note above the table)</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;keyword&gt;</td>
<td>N/A</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

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

7 The block vs. inline designation for the `<foreign>` element is likely to change for some specializations.
<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;keywords&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;li&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lines&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitleFooNote</td>
</tr>
<tr>
<td>&lt;link&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;linkinfo&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;linklist&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitleFooNote</td>
</tr>
<tr>
<td>&lt;linkpool&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;linktext&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lq&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@reftitle</td>
</tr>
<tr>
<td>&lt;metadata&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;navtitle&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;no-topic-nesting&gt;</td>
<td>N/A</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>&lt;note&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@othertype, @spectitleFooNote</td>
</tr>
<tr>
<td>&lt;object&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@standby</td>
</tr>
<tr>
<td>&lt;ol&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitleFooNote</td>
</tr>
<tr>
<td>&lt;othermeta&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td>@content</td>
</tr>
<tr>
<td>&lt;p&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;param&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>&lt;permissions&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;ph&gt;</td>
<td>N/A</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;platform&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;pre&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitleFooNote</td>
</tr>
<tr>
<td>&lt;prodinfo&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;prodname&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;prognum&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;prolog&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;publisher&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;q&gt;</td>
<td>N/A</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;related-links&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>------------------</td>
<td>------------------------------</td>
<td>-----------------------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>&lt;required-cleanup&gt;</td>
<td>N/A</td>
<td>block***Footnote.</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;resourceid&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;revised&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;row&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;searchtitle&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;section&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>&amp;@spectitleFootnote.</td>
</tr>
<tr>
<td>&lt;sectiondiv&gt; (new in DITA 1.2)</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;series&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;shortdesc&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;simpletable&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>&amp;@spectitleFootnote.</td>
</tr>
<tr>
<td>&lt;sl&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>&amp;@spectitleFootnote.</td>
</tr>
<tr>
<td>&lt;sli&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;source&gt;</td>
<td>N/A</td>
<td>block (metadata)</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;state&gt;</td>
<td>N/A</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td>&amp;@value</td>
</tr>
<tr>
<td>&lt;stentry&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>&amp;@specentryFootnote.</td>
</tr>
<tr>
<td>&lt;sthead&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;strow&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;table&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;tbody&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;term&gt;</td>
<td>N/A</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;text&gt; (new in DITA 1.2)</td>
<td>N/A</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;tgroup&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;thead&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;title&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;titlealts&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;tm&gt;</td>
<td>N/A</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;topic&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>&amp;@spectitleFootnote.</td>
</tr>
<tr>
<td>&lt;ul&gt;</td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>&amp;@spectitleFootnote.</td>
</tr>
</tbody>
</table>
### map elements

As of DITA 1.2, the `@navtitle` attribute is deprecated, for translation purposes, in favor of the new `<navtitle>` element. The `<navtitle>` element is also available in topics, and is included in the topic table above.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;anchor&gt;</code></td>
<td>N/A</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td><code>&lt;linktext&gt;</code></td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;map&gt;</code></td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@title&lt;sup&gt;8&lt;/sup&gt;</td>
</tr>
<tr>
<td><code>&lt;navref&gt;</code></td>
<td>N/A</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td><code>&lt;relcell&gt;</code></td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;relcolspec&gt;</code></td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;relheader&gt;</code></td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;relrow&gt;</code></td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;reltable&gt;</code></td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;searchtitle&gt;</code></td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;shortdesc&gt;</code></td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;topicmeta&gt;</code></td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;topicref&gt;</code></td>
<td>N/A</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;ux-window&gt;</code></td>
<td>N/A</td>
<td>N/A (empty)</td>
<td>N/A (empty)</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

### bookmap elements (new in DITA 1.1)

The bookmap specialization contains many phrase-based elements inside the bookmeta. These are metadata, and should not be translated.

As of DITA 1.2, the `@navtitle` attribute is deprecated, for translation purposes, in favor of the new `<navtitle>` element.

---

<sup>8</sup> (deprecated as of DITA 1.2 for translation purposes)
<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;abbrevlist&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td><code>&lt;amendments&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td><code>&lt;appendix&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td><code>&lt;approved&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;backmatter&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td><code>&lt;bibliolist&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td><code>&lt;bookabstract&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td><code>&lt;bookchangehistory&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;bookevent&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;bookeventtype&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;bookid&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;booklibrary&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;booklist&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td><code>&lt;booklists&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;bookmap&gt;</code></td>
<td><code>&lt;map&gt;</code></td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>removed title attribute</td>
</tr>
<tr>
<td><code>&lt;bookmeta&gt;</code></td>
<td><code>&lt;topicmeta</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;booknumber&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;bookowner&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;bookpartno&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;bookrestriction&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;bookrights&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;booktitle&gt;</code></td>
<td><code>&lt;title&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;booktitlealt&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;chapter&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td><code>&lt;colophon&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td><code>&lt;completed&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;copyrfirst&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;copyrlast&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;day&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;dedication&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td><code>&lt;draftintro&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------</td>
<td>------------------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>&lt;edited&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;edition&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;figurelist&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;frontmatter&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;glossarylist&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;indexlist&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;isbn&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;mainbooktitle&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;maintainer&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;month&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;notices&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;organization&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;part&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;person&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;preface&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;printlocation&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;published&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;publisherinformation&gt;</td>
<td>&lt;publisher&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;publishtype&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;reviewed&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;revisionid&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;started&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;summary&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;tablelist&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;tested&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;toc&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;trademarklist&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;volume&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;year&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>
### Concept Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Specialized From</th>
<th>Inherits Everything From Ancestor?</th>
<th>Block/Inline (Presentation)</th>
<th>Block/Inline (Translation)</th>
<th>Translatable Content?</th>
<th>Translatable Attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;conbody&gt;</code></td>
<td><code>&lt;body&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;conbodydiv&gt;</code></td>
<td><code>&lt;bodydiv&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;concept&gt;</code></td>
<td><code>&lt;topic&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

### Glossentry Elements (new in DITA 1.1)

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Specialized From</th>
<th>Inherits Everything From Ancestor?</th>
<th>Block/Inline (Presentation)</th>
<th>Block/Inline (Translation)</th>
<th>Translatable Content?</th>
<th>Translatable Attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;glossAbbreviation&gt;</code></td>
<td><code>&lt;title&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossAcronym&gt;</code></td>
<td><code>&lt;title&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossAlt&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>removes spectitle</td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossAlternateFor&gt;</code></td>
<td><code>&lt;xref&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>n/a (empty element)</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossbody&gt;</code></td>
<td><code>&lt;body&gt;, </code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td><code>&lt;conbody&gt;</code></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossdef&gt;</code></td>
<td><code>&lt;abstract&gt;, </code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossentry&gt;</code></td>
<td><code>&lt;topic&gt;, </code>&lt;concept&gt;`</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossPartOfSpeech&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossProperty&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossScopeNote&gt;</code></td>
<td><code>&lt;note&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@othertype</td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossShortForm&gt;</code></td>
<td><code>&lt;title&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossStatus&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossSurfaceForm&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;glossSymbol&gt;</code></td>
<td><code>&lt;image&gt;</code></td>
<td>yes</td>
<td>block when @placement=break,</td>
<td>block when @placement=break,</td>
<td>yes</td>
<td>removes @alt</td>
</tr>
<tr>
<td>(new in DITA 1.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>-------------------------------------</td>
<td>----------------------------</td>
<td>---------------------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>&lt;glossSynonym&gt;</td>
<td>&lt;title&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;glossterm&gt;</td>
<td>&lt;title&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;glossUsage&gt;</td>
<td>&lt;note&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@othertype (removes @spectitle)</td>
</tr>
</tbody>
</table>

**reference elements**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;propdesc&gt;</td>
<td>&lt;stentry&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@specentry Footnote.</td>
</tr>
<tr>
<td>&lt;propdeschd&gt;</td>
<td>&lt;stentry&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@specentry Footnote.</td>
</tr>
<tr>
<td>&lt;properties&gt;</td>
<td>&lt;simpletable&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td>&lt;property&gt;</td>
<td>&lt;strow&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;prophed&gt;</td>
<td>&lt;sthead&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;proptype&gt;</td>
<td>&lt;stentry&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@specentry Footnote.</td>
</tr>
<tr>
<td>&lt;proptypehd&gt;</td>
<td>&lt;stentry&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@specentry Footnote.</td>
</tr>
<tr>
<td>&lt;propvalue&gt;</td>
<td>&lt;stentry&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@specentry Footnote.</td>
</tr>
<tr>
<td>&lt;propvaluehd&gt;</td>
<td>&lt;stentry&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@specentry Footnote.</td>
</tr>
<tr>
<td>&lt;refbody&gt;</td>
<td>&lt;body&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@specentry Footnote.</td>
</tr>
<tr>
<td>&lt;refbodydiv&gt;</td>
<td>&lt;bodydiv&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;reference&gt;</td>
<td>&lt;topic&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;refsyn&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
</tbody>
</table>
## Task Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;chdesc&gt;</code></td>
<td><code>&lt;stentry&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@specentry</code> Footnote.</td>
</tr>
<tr>
<td><code>&lt;chdescd&gt;</code></td>
<td><code>&lt;stentry&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@specentry</code> Footnote.</td>
</tr>
<tr>
<td><code>&lt;chhead&gt;</code></td>
<td><code>&lt;sthead&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td></td>
<td><code>@specentry</code> Footnote.</td>
</tr>
<tr>
<td><code>&lt;choice&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;choices&gt;</code></td>
<td><code>&lt;ul&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes @spectitle</code></td>
</tr>
<tr>
<td><code>&lt;choicetable&gt;</code></td>
<td><code>&lt;simpletable&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@spectitle</code> Footnote.</td>
</tr>
<tr>
<td><code>&lt;choption&gt;</code></td>
<td><code>&lt;stentry&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@specentry</code> Footnote.</td>
</tr>
<tr>
<td><code>&lt;choptionhd&gt;</code></td>
<td><code>&lt;stentry&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@specentry</code> Footnote.</td>
</tr>
<tr>
<td><code>&lt;chrow&gt;</code></td>
<td><code>&lt;strow&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;cmd&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>NO</td>
<td>inline</td>
<td>block</td>
<td>yes</td>
<td><code>removes @spectitle</code></td>
</tr>
<tr>
<td><code>&lt;context&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes @spectitle</code></td>
</tr>
<tr>
<td><code>&lt;info&gt;</code></td>
<td><code>&lt;itemgroup&gt;</code></td>
<td>NO</td>
<td>inline</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;postreq&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes @spectitle</code></td>
</tr>
<tr>
<td><code>&lt;prereq&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes @spectitle</code></td>
</tr>
<tr>
<td><code>&lt;result&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes @spectitle</code></td>
</tr>
<tr>
<td><code>&lt;step&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;stepresult&gt;</code></td>
<td><code>&lt;itemgroup&gt;</code></td>
<td>NO</td>
<td>inline</td>
<td>block</td>
<td>yes</td>
<td><code>removes @spectitle</code></td>
</tr>
<tr>
<td><code>&lt;steps&gt;</code></td>
<td><code>&lt;ol&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes @spectitle</code></td>
</tr>
<tr>
<td><code>&lt;steps-informal&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes @spectitle</code></td>
</tr>
<tr>
<td><code>&lt;steps-unordered&gt;</code></td>
<td><code>&lt;ul&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes @spectitle</code></td>
</tr>
<tr>
<td><code>&lt;stepsection&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes @spectitle</code></td>
</tr>
<tr>
<td><code>&lt;steptroubleshooting&gt;</code></td>
<td><code>&lt;itemgroup&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------</td>
<td>-------------------------------------</td>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>&lt;stepxmp&gt;</td>
<td>&lt;itemgroup&gt;</td>
<td>NO</td>
<td>inline</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;task&gt;</td>
<td>&lt;topic&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;taskbody&gt;</td>
<td>&lt;body&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;tasktroubleshooting&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>removes @specentry</td>
</tr>
<tr>
<td>&lt;tutorialinfo&gt;</td>
<td>&lt;itemgroup&gt;</td>
<td>NO</td>
<td>inline</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

**troubleshooting elements (new in DITA 1.3)**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cause&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@specentry</td>
</tr>
<tr>
<td>&lt;condition&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@specentry</td>
</tr>
<tr>
<td>&lt;remedy&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@specentry</td>
</tr>
<tr>
<td>&lt;responsibleParty&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;troublebody&gt;</td>
<td>&lt;body&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;troubleshooting&gt;</td>
<td>&lt;topic&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;troubleSolution&gt;</td>
<td>&lt;bodydiv&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

**hi-d elements (highlight domain)**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;b&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;line-through&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td>(new in DITA 1.3)</td>
</tr>
<tr>
<td>&lt;i&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;overline&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td>(new in DITA 1.3)</td>
</tr>
<tr>
<td>&lt;sub&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>
### Indexing-D Elements (Indexing Domain) *(New in DITA 1.1)*

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (Presentation)</th>
<th>Block/Inline (Translation)</th>
<th>Translatable Content?</th>
<th>Translatable Attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;index-see&gt;</code></td>
<td><code>&lt;index-base&gt;</code></td>
<td>yes</td>
<td>block***Footnote.*</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;index-see-also&gt;</code></td>
<td><code>&lt;index-base&gt;</code></td>
<td>yes</td>
<td>block***Footnote.*</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;index-sort-as&gt;</code></td>
<td><code>&lt;index-base&gt;</code></td>
<td>yes</td>
<td>block***Footnote.*</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

### PR-D Elements (Programming Domain)

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (Presentation)</th>
<th>Block/Inline (Translation)</th>
<th>Translatable Content?</th>
<th>Translatable Attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;apiname&gt;</code></td>
<td><code>&lt;keyword&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td>@spectitleF</td>
</tr>
<tr>
<td><code>&lt;coderef&gt;</code></td>
<td><code>&lt;xref&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>N/A (empty element)</td>
<td>N/A</td>
<td>ootnote.</td>
</tr>
<tr>
<td><code>&lt;codeph&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;delim&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;fragment&gt;</code></td>
<td><code>&lt;figgroup&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;fragref&gt;</code></td>
<td><code>&lt;xref&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;groupchoice&gt;</code></td>
<td><code>&lt;figgroup&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;groupcomp&gt;</code></td>
<td><code>&lt;figgroup&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;kwd&gt;</code></td>
<td><code>&lt;keyword&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;oper&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;option&gt;</code></td>
<td><code>&lt;keyword&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>
### sw-d elements (software domain)

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cmdname&gt;</td>
<td>&lt;keyword&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;filepath&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;msgblock&gt;</td>
<td>&lt;pre&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;msgnum&gt;</td>
<td>&lt;keyword&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;msgph&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;systemoutput&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;userinput&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;varname&gt;</td>
<td>&lt;keyword&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

---

Footnote.
### ui-d elements (UI domain)

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;menucascade&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;screen&gt;</td>
<td>&lt;pre&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle</td>
</tr>
<tr>
<td>&lt;shortcut&gt;</td>
<td>&lt;keyword&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;uicontrol&gt;</td>
<td>&lt;ph&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;wintitle&gt;</td>
<td>&lt;keyword&gt;</td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

### ut-d elements (Utilities domain)

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;area&gt;</td>
<td>&lt;figgroup&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;coords&gt;</td>
<td>&lt;ph&gt;</td>
<td>NO</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;imagemap&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td>@spectitle</td>
</tr>
<tr>
<td>&lt;shape&gt;</td>
<td>&lt;keyword&gt;</td>
<td>NO</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;sort-as&gt;</td>
<td>&lt;data&gt;</td>
<td>NO</td>
<td>block***Footnote.</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

### mapgroup-d elements (mapgroup domain)

As of DITA 1.2, the @navtitle attribute is deprecated, for translation purposes, in favor of the new <navtitle> element.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;anchorref&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;keydef&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;mapref&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;topicgroup&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------</td>
<td>------------------------------------</td>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>&lt;topichread&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;topicset&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;topicsetref&gt;</td>
<td>(new in DITA 1.2)</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
</tbody>
</table>

### ditavalref-d elements (ditavalref domain) (new in DITA 1.3)

The @navtitle attribute is deprecated, for translation purposes, in favor of the `<navtitle>` element.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ditavalmeta&gt;</td>
<td>&lt;topicmeta&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;ditavalref&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;dvrKeyscopePrefix&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;dvrKeyscopeSuffix&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;dvrResourcePrefix&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;dvrResourceSuffix&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

### xnal-d elements (XNAL domain) (new in DITA 1.1)

The XNAL information is all metadata, so it generally does not need to be translated. Exceptions might be needed when selections from this metadata are used for display purposes. The standard behavior might need to change based on local business rules. For example, in some cases it might be appropriate to translate the honorific, country, or organizationname elements.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;addressdetails&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;administrativearea&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;authorinformation&gt;</td>
<td>&lt;author&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;contactnumber&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;contactnumbers&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------</td>
<td>------------------------------------</td>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>&lt;country&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;emailaddress&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;emailaddresses&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;firstname&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;generationidentifier&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;honorific&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;lastname&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;locality&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;localityname&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;middlename&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;namedetails&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;organizationinfo&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;organizationname&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;organizationnamedetails&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;otherinfo&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;personinfo&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;personname&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;postalcode&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;thoroughfare&gt;</td>
<td>&lt;ph&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;url&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;urls&gt;</td>
<td>&lt;data&gt;</td>
<td>no</td>
<td>block</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

**Delayed Conref Resolution domain elements (new in DITA 1.2)**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;anchorid&gt;</td>
<td>&lt;keyword&gt;</td>
<td>no</td>
<td>N/A (metadata)</td>
<td>n/a</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;anchorkey&gt;</td>
<td>&lt;keyword&gt;</td>
<td>no</td>
<td>N/A (metadata)</td>
<td>n/a</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;exportanchors&gt;</td>
<td>&lt;keywords&gt;</td>
<td>no</td>
<td>N/A (metadata)</td>
<td>n/a</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>
## Hazard Statement Domain *(new in DITA 1.2)*

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;consequence&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;hazardstatement&gt;</code></td>
<td><code>&lt;note&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@othertype, @spectitle</td>
</tr>
<tr>
<td><code>&lt;hazardsymbol&gt;</code></td>
<td><code>&lt;image&gt;</code></td>
<td>yes</td>
<td>block when @placement= break, otherwise inline</td>
<td>block when @placement= break, otherwise inline</td>
<td>yes</td>
<td>removes @alt</td>
</tr>
<tr>
<td><code>&lt;howtoavoid&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;messagepanel&gt;</code></td>
<td><code>&lt;ul&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Francisco</td>
</tr>
<tr>
<td><code>&lt;typeofhazard&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

### glossgroup elements *(new in DITA 1.2)*

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;glossgroup&gt;</code></td>
<td><code>&lt;topic&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

### glossref domain elements *(new in DITA 1.2)*

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;glossref&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
</tbody>
</table>

### abbreviated-form domain elements *(new in DITA 1.2)*

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;abbreviated-form&gt;</code></td>
<td><code>&lt;term&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>N/A empty element</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>
## task requirements domain *(new in DITA 1.2)*

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;closereqs&gt;</code></td>
<td><code>&lt;section&gt;</code>, <code>&lt;postreq&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;esttime&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;noconds&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;nosafety&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;nospare&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;nosupe&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;nosupply&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;perscat&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;perskill&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;personnel&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;prelreq&gt;</code></td>
<td><code>&lt;section&gt;</code>, <code>&lt;prereq&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;reqcond&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;reqconda&gt;</code></td>
<td><code>&lt;ol&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;reqcontp&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;reqpers&gt;</code></td>
<td><code>&lt;ol&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;safecond&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;safety&gt;</code></td>
<td><code>&lt;ol&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;spare&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;spares&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;sparesli&gt;</code></td>
<td><code>&lt;ul&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;supeqli&gt;</code></td>
<td><code>&lt;ul&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;supequi&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;supequip&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;supplies&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;supply&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;supplyli&gt;</code></td>
<td><code>&lt;ul&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>
### markup-d elements (markup domain) (new in DITA 1.3)

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;markupname&gt;</td>
<td>&lt;keyword&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

### xml-d elements (markup domain) (new in DITA 1.3)

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numcharref&gt;</td>
<td>&lt;keyword&gt;, &lt;markupname&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;parameterentity&gt;</td>
<td>&lt;keyword&gt;, &lt;markupname&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;textentity&gt;</td>
<td>&lt;keyword&gt;, &lt;markupname&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;xmlatt&gt;</td>
<td>&lt;keyword&gt;, &lt;markupname&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;xmlelement&gt;</td>
<td>&lt;keyword&gt;, &lt;markupname&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;xmlnsname&gt;</td>
<td>&lt;keyword&gt;, &lt;markupname&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;xmlpi&gt;</td>
<td>&lt;keyword&gt;, &lt;markupname&gt;</td>
<td>no</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

### relmgmt-d elements (release management domain) (new in DITA 1.3)

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;change-historylist&gt;</td>
<td>&lt;metadata&gt;</td>
<td>NO</td>
<td>N/A (metadata)</td>
<td>N/A</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;change-completed&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>N/A</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;change-item&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>N/A</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;change-person&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>N/A</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;change-organization&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>N/A</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;change-revisionid&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>N/A</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------</td>
<td>------------------------------------</td>
<td>----------------------------</td>
<td>-----------------------------</td>
<td>------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td><code>&lt;change-request-reference&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>N/A</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;change-request-system&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>N/A</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;change-request-id&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>N/A</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;change-started&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>N/A</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;change-summary&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>N/A</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

**equation-d elements (equation domain) (new in DITA 1.3)**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;equation-inline&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;equation-block&gt;</code></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;equation-number&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;equation-figure&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

**mathml-d elements (MathML domain) (new in DITA 1.3)**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;mathml&gt;</code></td>
<td><code>&lt;foreign&gt;</code></td>
<td>yes</td>
<td>Variable: depends on math content</td>
<td>Depends on math content</td>
<td>Follow rules established for MathML standard</td>
<td></td>
</tr>
<tr>
<td><code>&lt;mathmlref&gt;</code></td>
<td><code>&lt;xref&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>N/A (empty)</td>
<td>no (empty)</td>
<td></td>
</tr>
</tbody>
</table>
### svg-d elements (SVG domain) (new in DITA 1.3)

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;svg-container&gt;</td>
<td>&lt;foreign&gt;</td>
<td>yes</td>
<td>Variable: depends on SVG content</td>
<td>Depends on SVG content</td>
<td>Follow rules established for SVG standard</td>
<td></td>
</tr>
<tr>
<td>&lt;svgref&gt;</td>
<td>&lt;xref&gt;</td>
<td>yes</td>
<td>inline</td>
<td>N/A (empty)</td>
<td>no (empty)</td>
<td></td>
</tr>
</tbody>
</table>

### classification domain elements (new in DITA 1.2)

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;subjectCell&gt;</td>
<td>&lt;relcell&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;subjectref&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;topicapply&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;topicCell&gt;</td>
<td>&lt;relcell&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;topicsubject&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;topicSubjectHeader&gt;</td>
<td>&lt;relrow&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;topicSubjectRow&gt;</td>
<td>&lt;relrow&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;topicSubjectTable&gt;</td>
<td>&lt;reltable&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

### subject scheme elements (new in DITA 1.2)

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;attributedef&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;defaultSubject&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;elementdef&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A (metadata)</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;enumerationdef&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;hasInstance&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;hasKind&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;hasNarrower&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;hasPart&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;hasRelated&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------</td>
<td>-------------------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>&lt;relatedSubjects&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;subjectdef&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;subjectHead&gt;</td>
<td>&lt;topicref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;subjectHeadMeta&gt;</td>
<td>&lt;topicmeta&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;schemeref&gt;</td>
<td>&lt;relref&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@navtitle</td>
</tr>
<tr>
<td>&lt;subjectRel&gt;</td>
<td>&lt;relrow&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;subjectRelHeader&gt;</td>
<td>&lt;relrow&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;subjectRelTable&gt;</td>
<td>&lt;reltable&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;subjectRole&gt;</td>
<td>&lt;relcell&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;subjectScheme&gt;</td>
<td>&lt;map&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>removes @title</td>
</tr>
</tbody>
</table>

**learningAssessment elements (new in DITA 1.2)**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;learningAssessment&gt;</td>
<td>&lt;topic&gt;, &lt;learningBase&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;learningAssessmentbody&gt;</td>
<td>&lt;body&gt;, &lt;learningBasebody&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

**learning base elements (new in DITA 1.2, two elements added in DITA 1.3)**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;lcAudience&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td>&lt;lcChallenge&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td>&lt;lcDuration&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td>&lt;lcInstruction&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td>&lt;lcInteraction&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------</td>
<td>-------------------------------------</td>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td><code>&lt;lcInteractionBase&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle</td>
</tr>
<tr>
<td><code>&lt;lcInteractionBase2&gt;</code></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcInteractionLabel2&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>(new in DITA 1.3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcIntro&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle</td>
</tr>
<tr>
<td><code>&lt;lcNextSteps&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle</td>
</tr>
<tr>
<td><code>&lt;lcObjective&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcObjectives&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle</td>
</tr>
<tr>
<td><code>&lt;lcObjectivesGroup&gt;</code></td>
<td><code>&lt;ul&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcObjectivesStem&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcPrereqs&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle</td>
</tr>
<tr>
<td><code>&lt;lcResources&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle</td>
</tr>
<tr>
<td><code>&lt;lcReview&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle</td>
</tr>
<tr>
<td><code>&lt;lcSummary&gt;</code></td>
<td><code>&lt;section&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle</td>
</tr>
<tr>
<td><code>&lt;lcTime&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td>@spectitle</td>
</tr>
<tr>
<td><code>&lt;learningBase&gt;</code></td>
<td><code>&lt;topic&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;learningBasebody&gt;</code></td>
<td><code>&lt;body&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

**learningContent elements (new in DITA 1.2)**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;learningContent&gt;</code></td>
<td><code>&lt;topic&gt;, &lt;learningBase&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;learningContentbody&gt;</code></td>
<td><code>&lt;body&gt;, &lt;learningBasebody&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>
### learningOverview elements *(new in DITA 1.2)*

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;learningOverview&gt;</code></td>
<td><code>&lt;topic&gt;, &lt;learningBase&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;learningOverviewbody&gt;</code></td>
<td><code>&lt;body&gt;, &lt;learningBasebody&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

### learningSummary elements *(new in DITA 1.2)*

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;learningSummary&gt;</code></td>
<td><code>&lt;topic&gt;, &lt;learningBase&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;learningSummarybody&gt;</code></td>
<td><code>&lt;body&gt;, &lt;learningBasebody&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

### learningPlan elements *(new in DITA 1.2)*

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;lcAge&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcAssessment&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcAttitude&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcBackground&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcCIN&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td><code>&lt;lcClassroom&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td><code>&lt;lcClient&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td><code>&lt;lcConstraints&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td><code>&lt;lcDelivDate&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td><code>&lt;lcDelivery&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcDownloadTime&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>@spectitle Footnote.</td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------</td>
<td>------------------------------------</td>
<td>----------------------------</td>
<td>---------------------------</td>
<td>-----------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>&lt;lcEdLevel&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcFileSizeLimitations&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcGapAnalysis&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcGapItem&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcGapItemDelta&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcGeneralDescription&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcGoals&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcGraphics&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcHandouts&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcIntervention&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcInterventionItem&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcJtaItem&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcKnowledge&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcLearnStrat&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcLMS&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcModDate&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcMotivation&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcNeeds&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcNeedsAnalysis&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcNoLMS&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcOJT&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcOrganizational&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>&lt;lcOrgConstraints&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------</td>
<td>-------------------------------------</td>
<td>----------------------------</td>
<td>---------------------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>&lt;lcPlanAudience&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcPlanDescrip&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcPlanObjective&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcPlanPrereqs&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcPlanResources&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcPlanSubject&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcPlanTitle&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcPlayers&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcProcesses&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcProject&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcResolution&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcSecurity&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcSkills&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcSpecChars&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcTask&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcTaskItem&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcTechnical&gt;</td>
<td>&lt;section&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcValues&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcViewers&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcW3C&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcWorkEnv&gt;</td>
<td>&lt;fig&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;lcWorkEnvDescription&gt;</td>
<td>&lt;p&gt;</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>&lt;learningPlan&gt;</td>
<td>&lt;topic&gt;,</td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td> @spectitle</td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------</td>
<td>------------------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td><code>&lt;learningPlanbody&gt;</code></td>
<td><code>&lt;body&gt;</code>, <code>&lt;learningBasebody&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

**learning maps (new in DITA 1.3)**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;learningGroupMap&gt;</code></td>
<td><code>&lt;map&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes @title</code></td>
</tr>
<tr>
<td><code>&lt;learningObjectMap&gt;</code></td>
<td><code>&lt;map&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes @title</code></td>
</tr>
</tbody>
</table>

**learning map domain elements (new in DITA 1.2)**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;learningContentRef&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@navtitle</code></td>
</tr>
<tr>
<td><code>&lt;learningGroup&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@navtitle</code></td>
</tr>
<tr>
<td><code>&lt;learningGroupMapRef&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@navtitle</code></td>
</tr>
<tr>
<td><code>&lt;learningObject&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@navtitle</code></td>
</tr>
<tr>
<td><code>&lt;learningObjectMapRef&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@navtitle</code></td>
</tr>
<tr>
<td><code>&lt;learningOverviewRef&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@navtitle</code></td>
</tr>
<tr>
<td><code>&lt;learningPlanRef&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@navtitle</code></td>
</tr>
<tr>
<td><code>&lt;learningPostAssessmentRef&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@navtitle</code></td>
</tr>
<tr>
<td><code>&lt;learningPreAssessmentRef&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@navtitle</code></td>
</tr>
<tr>
<td><code>&lt;learningSummaryRef&gt;</code></td>
<td><code>&lt;topicref&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@navtitle</code></td>
</tr>
</tbody>
</table>

**learning domain elements (new in DITA 1.2, updated in DITA 1.3)**

DITA 1.3 adds a copy of the learning domain with 2 added to each element type name. The new elements have the same translation properties as the originals, but usually have different and more appropriate specialization ancestors.
<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;lcAnswerContent&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcAnswerContent2&gt;</code></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>(New in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcAnswerOption&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcAnswerOption2&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>(New in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcAnswerOptionGroup&gt;</code></td>
<td><code>&lt;ul&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcAnswerOptionGroup2&gt;</code></td>
<td><code>&lt;ul&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>(New in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcArea&gt;</code></td>
<td><code>&lt;figgroup&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcArea2&gt;</code></td>
<td><code>&lt;figgroup&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>(new in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcAreaCoords&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>NO</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcAreaCoords2&gt;</code></td>
<td><code>&lt;ph&gt;</code></td>
<td>NO</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td>(new in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcAreaShape&gt;</code></td>
<td><code>&lt;keyword&gt;</code></td>
<td>NO</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcAreaShape2&gt;</code></td>
<td><code>&lt;keyword&gt;</code></td>
<td>NO</td>
<td>inline</td>
<td>inline</td>
<td>no</td>
<td>(new in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcAsset&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcAsset2&gt;</code></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>(New in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcCorrectResponse&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcCorrectResponse2&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td>(New in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcFeedback&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcFeedback2&gt;</code></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>(New in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcFeedbackCorrect&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcFeedbackCorrect2&gt;</code></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>(New in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcFeedbackIncorrect&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcFeedbackIncorrect2&gt;</code></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>(New in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcHotspot&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcHotspot2&gt;</code></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td>(New in DITA 1.3)</td>
</tr>
<tr>
<td><code>&lt;lcHotspotMap&gt;</code></td>
<td><code>&lt;figgroup&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------</td>
<td>-------------------------------------</td>
<td>----------------------------</td>
<td>----------------------------</td>
<td>------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td><code>&lt;lcHotspotMap2&gt;</code> (<em>new in DITA 1.3</em>)</td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@spectitle</code></td>
</tr>
<tr>
<td><code>&lt;lcInstructornote&gt;</code></td>
<td><code>&lt;note&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@spectitle</code></td>
</tr>
<tr>
<td><code>&lt;lcInstructornote2&gt;</code> (<em>new in DITA 1.3</em>)</td>
<td><code>&lt;note&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>@spectitle</code></td>
</tr>
<tr>
<td><code>&lt;lcItem&gt;</code></td>
<td><code>&lt;stentry&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes</code> @specentry</td>
</tr>
<tr>
<td><code>&lt;lcItem2&gt;</code> (<em>New in DITA 1.3</em>)</td>
<td><code>&lt;stentry&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes</code> @specentry</td>
</tr>
<tr>
<td><code>&lt;lcMatching&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes</code> @specentry</td>
</tr>
<tr>
<td><code>&lt;lcMatching2&gt;</code> (<em>New in DITA 1.3</em>)</td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcMatchingItem&gt;</code></td>
<td><code>&lt;stentry&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes</code> @specentry</td>
</tr>
<tr>
<td><code>&lt;lcMatchingItem2&gt;</code> (<em>New in DITA 1.3</em>)</td>
<td><code>&lt;stentry&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes</code> @specentry</td>
</tr>
<tr>
<td><code>&lt;lcMatchingItemFeedback&gt;</code></td>
<td><code>&lt;stentry&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes</code> @specentry</td>
</tr>
<tr>
<td><code>&lt;lcMatchingItemFeedback2&gt;</code> (<em>New in DITA 1.3</em>)</td>
<td><code>&lt;stentry&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes</code> @specentry</td>
</tr>
<tr>
<td><code>&lt;lcMatchTable&gt;</code></td>
<td><code>&lt;simpletable&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes</code> @spectitle</td>
</tr>
<tr>
<td><code>&lt;lcMatchTable2&gt;</code> (<em>New in DITA 1.3</em>)</td>
<td><code>&lt;simpletable&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes</code> @spectitle</td>
</tr>
<tr>
<td><code>&lt;lcMatchingHeader&gt;</code></td>
<td><code>&lt;sthead&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td><code>removes</code> @spectitle</td>
</tr>
<tr>
<td><code>&lt;lcMatchingHeader2&gt;</code> (<em>New in DITA 1.3</em>)</td>
<td><code>&lt;sthead&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcMatchingPair&gt;</code></td>
<td><code>&lt;strow&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcMatchingPair2&gt;</code> (<em>New in DITA 1.3</em>)</td>
<td><code>&lt;strow&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcMultipleSelect&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcMultipleSelect2&gt;</code> (<em>new in DITA 1.3</em>)</td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcOpenAnswer&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcOpenAnswer2&gt;</code> (<em>New in DITA 1.3</em>)</td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcOpenQuestion&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------------------</td>
<td>------------------------------------</td>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>----------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td><code>&lt;lcOpenQuestion2&gt;</code> <em>(New in DITA 1.3)</em></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcQuestion&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcQuestion2&gt;</code> <em>(New in DITA 1.3)</em></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcQuestionBase&gt;</code></td>
<td><code>&lt;p&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcQuestionBase2&gt;</code> <em>(New in DITA 1.3)</em></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcSequence&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcSequence2&gt;</code> <em>(New in DITA 1.3)</em></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcSequenceOption&gt;</code></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcSequenceOption2&gt;</code> <em>(New in DITA 1.3)</em></td>
<td><code>&lt;li&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcSequenceOptionGroup&gt;</code></td>
<td><code>&lt;ol&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcSequenceOptionGroup2&gt;</code> <em>(New in DITA 1.3)</em></td>
<td><code>&lt;ol&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcSequencing&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcSequencing2&gt;</code> <em>(new in DITA 1.3)</em></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcSingleSelect&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcSingleSelect2&gt;</code> <em>(new in DITA 1.3)</em></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcTrueFalse&gt;</code></td>
<td><code>&lt;fig&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcTrueFalse2&gt;</code> <em>(new in DITA 1.3)</em></td>
<td><code>&lt;div&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

**learning metadata domain (new in DITA 1.2)**

<table>
<thead>
<tr>
<th>Element name</th>
<th>Specialized from</th>
<th>Inherits everything from ancestor?</th>
<th>Block/Inline (presentation)</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
<th>Translatable attributes?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;lcLom&gt;</code></td>
<td><code>&lt;metadata&gt;</code></td>
<td>yes</td>
<td>block</td>
<td>block</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcLomAggregationLevel&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcLomContext&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcLomCoverage&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><code>&lt;lcLomDifficulty&gt;</code></td>
<td><code>&lt;data&gt;</code></td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Element name</td>
<td>Specialized from</td>
<td>Inherits everything from ancestor?</td>
<td>Block/Inline (presentation)</td>
<td>Block/Inline (translation)</td>
<td>Translatable content?</td>
<td>Translatable attributes?</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>------------------</td>
<td>-------------------------------------</td>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>&lt;lomInstallationRemarks&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;lomIntendedUserRole&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;lomInteractivityLevel&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;lomInteractivityType&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;lomLearningResourceType&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;lomOtherPlatformRequirements&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;lomSemanticDensity&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;lomStructure&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;lomTechRequirement&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;lomTypicalAgeRange&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>&lt;lomTypicalLearningTime&gt;</td>
<td>&lt;data&gt;</td>
<td>yes</td>
<td>N/A</td>
<td>block</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

**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.

As of DITA 1.2, the only element that directly contains text for translation is `<alt-text>`.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Block/Inline (translation)</th>
<th>Translatable content?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;alt-text&gt;</td>
<td>block</td>
<td>yes (inside nested elements)</td>
</tr>
<tr>
<td>&lt;endflag&gt;</td>
<td>block</td>
<td>yes (inside nested elements)</td>
</tr>
<tr>
<td>&lt;prop&gt;</td>
<td>block</td>
<td>yes (inside nested elements)</td>
</tr>
<tr>
<td>&lt;revprop&gt;</td>
<td>block</td>
<td>yes (inside nested elements)</td>
</tr>
<tr>
<td>&lt;startflag&gt;</td>
<td>block</td>
<td>yes (inside nested elements)</td>
</tr>
<tr>
<td>&lt;style-conflict&gt;</td>
<td>block</td>
<td>N/A (empty element)</td>
</tr>
<tr>
<td>&lt;val&gt;</td>
<td>block</td>
<td>yes (inside nested elements)</td>
</tr>
</tbody>
</table>
# Appendix C Revision history

The following table contains information about revisions to this document.

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Editor</th>
<th>Description of changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>19 May 2019</td>
<td>Kristen James Eberlein</td>
<td>Generated working draft #01.Contains initial TOC for consideration.</td>
</tr>
<tr>
<td>02</td>
<td>24 May 2019</td>
<td>Kristen James Eberlein</td>
<td>Generated working draft #02. Contains reworked task element topics.</td>
</tr>
<tr>
<td>03</td>
<td>27 August 2017</td>
<td>Kristen James Eberlein</td>
<td>Generated working draft #03. Contains updated DITA source for the following proposals:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• #85: Several glossentry elements should allow <code>&lt;sub&gt;</code> and <code>&lt;sup&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• #106 Allow steps to nest</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Contains edited element-reference topics for the following domains:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Markup</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Programming (partial)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Software</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• User interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• XML mention</td>
</tr>
</tbody>
</table>