



## DITA Technical Committee

---

---

## Table of contents

DITA 2.0 glossary.....	3
------------------------	---

---

## DITA 2.0 glossary

Come and get it.

### <abstract>

The <abstract> element occurs between the topic title and the topic body. Abstract content is presented as the initial content of a topic. The <abstract> can contain paragraph-level content and one or more <shortdesc> elements, which can be used for providing link previews or summaries.

### addressing attribute

An attribute, such as @conref, @conkeyref, @keyref, and @href, that specifies an address.

### Alternate titles

Titles specifically for use in navigation or search. When not provided, the base title is used for all contexts.

### ancestor scope

A key scope that occurs any level above another key scope.

### <anchor>

The <anchor> element provides an integration point that another map can reference to insert its navigation into the referenced map's navigation tree. This element serves the same purpose as the <anchor> element in Eclipse help systems. The integration point might not be supported for all output formats.

### <anchorref>

Enables authors to define a map fragment that is pushed to the location defined by an anchor.

### Architectural attributes

The architectural attributes specify the version of DITA that the content supports; they also identify the DITA domains, structural types, and specializations that are in use by the content.

The architectural attributes should not be marked up in the source DITA map and topics. Instead, the values of the architectural attributes are handled by the processor when the content is processed, preferably through defaults set in the XML grammar. This practice ensures that the DITA content instances do not specify invalid values for the architectural attributes.

### attribute domain module

A domain module that defines a specialization of either the @base or @props attribute.

### attribute domain module

A vocabulary module that defines exactly one specialization of either the @base or @props attribute.

## attribute domain module

A domain module that defines a specialization of either the @base or @props attribute.

## attribute specialization

The DITA specialization mechanism provides a standard mechanism for defining that an attribute that is derived from an ancestor type. A specialized type inherits the semantics and default processing behavior from its ancestor type. Additional processing behavior can also be associated with the specialized descendant type.

## @audience

The intended audience of the content.

## @base

A generic attribute that has no specific purpose, but is intended to act as the basis for specialized attributes that have a simple value syntax like the conditional processing attributes (one or more alphanumeric values separated by whitespace or parenthesized groups of values).

## base element

## base type

An element or attribute type that is not a specialization. All base types are defined by the DITA specification.

## basic topic elements

Base topic elements include elements that make up the core building blocks of the DITA topic, such as <topic>, <body>, and <related-links>, as well as elements like <p> and <ph> that are used in many topic specializations. Some of these elements are also available inside the <topicmeta> map element.

## Block-level elements

Paragraphs, lists, figures, and tables are types of "block" elements. As a class of content, they can contain other blocks, phrases, or text, though the rules vary for each structure.

## Body

The topic body contains the topic content: paragraphs, lists, sections, and other content that the information type permits.

## <bodydiv>

The <bodydiv> element enables the arbitrary grouping of content within the body of a topic for the purpose of content reuse. The <bodydiv> element does not include a title. Content that requires a title should use <section> or <example>.

## body elements

body elements support the most common types of content authoring for topics: paragraphs, lists, phrases, figures, and other common types of exhibits in a document.

## 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. A `<bookmap>` consists 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.

## branch filtering

The branch filtering mechanism enables map authors to set filtering conditions for specific branches of a map. This mechanism makes it possible for multiple conditional-processing profiles to be applied within a single publication. Without the branch filtering mechanism, the conditions specified in a DITAVAL document are applied globally. With branch filtering, the `<ditavalref>` element specifies a DITAVAL document that can be applied to a subset of content; the location of the `<ditavalref>` element determines the content to which filtering conditions are applied. The filtering conditions then are used to filter the map branch itself (map elements used to create the branch), and the local maps or topics that are referenced by that branch.

## @cascade

Specifies whether the default rules for the cascading of metadata attributes in a DITA map apply. In addition to the following specified values, processors also MAY define additional values. @<>

### **cascade="merge"**

The metadata attributes cascade; the values of the metadata attributes are additive. This is the processing default for the `@cascade` attribute and was the only defined behavior for DITA 1.2 and earlier.

### **cascade="nomerge"**

The metadata attributes cascade; however, they are not additive for `<topicref>` elements that specify a different value for a specific metadata attribute.

If the cascading value for an attribute is already merged based on multiple ancestor elements, that merged value continues to cascade until a new value is encountered. That is, setting `cascade="nomerge"` does not undo merging that took place on ancestors.

For more information, see Example: How the cascade attribute functions.

## @cascade

The `@cascade` attribute can be used to modify the additive nature of attribute cascading (though it does not turn off cascading altogether). The attribute has two predefined values: "merge" and "nomerge".

### **- cascade="merge"**

The metadata attributes cascade; the values of the metadata attributes are additive. This is the processing default for the `@cascade` attribute and was the only defined behavior for DITA 1.2 and earlier.

### **- cascade="nomerge"**

The metadata attributes cascade; however, they are not additive for `<topicref>` elements that specify a different value for a specific metadata attribute. If the cascading value for an attribute is

already merged based on multiple ancestor elements, that merged value continues to cascade until a new value is encountered (that is, setting `cascade="nomerge"` does not undo merging that took place on ancestors).

<>

## Cascading

Certain map-level attributes cascade throughout a map, which facilitates attribute and metadata management. When attributes cascade, they apply to the elements that are children of the element where the attributes were specified. Cascading applies to a containment hierarchy, as opposed to an element-type hierarchy.

## child scope

A key scope that occurs directly within another key scope.

## @chunk

Specifies that the processor generates an interim set of DITA topics that are used as the input for the final processing. Using this attribute can produce the following output results:

- Multi-topic files are transformed into smaller files, for example, individual HTML files for each DITA topic
- Individual DITA topics are combined into a single file

Specifying a value for the `@chunk` attribute on a `<map>` element establishes chunking behavior that applies to the entire map, unless overridden by `@chunk` attributes that are set on more specific elements in the DITA map. For a detailed description of the `@chunk` attribute and its usage, see [Chunking](#).

## chunking

Content can be chunked (divided or merged into new output documents) in different ways for the purposes of delivering content and navigation. For example, content best authored as a set of separate topics might be delivered as a single Web page. A map author can use the `@chunk` attribute to split up multi-topic documents into component topics or to combine multiple topics into a single document as part of output processing.

## @class

This attribute identifies the specialization hierarchy for the element type. Every DITA element (except the `<dita>` element that is used as the root of a database document) MUST declare a `@class` attribute.

## class attribute syntax

The specialization hierarchy of each DITA element is declared as the value of the `@class` attribute. The `@class` attribute provides a mapping from the current name of the element to its more general equivalents, and also to more specialized equivalents. All specialization-aware processing can be defined in terms of `@class` attribute values. The `@class` attribute tells a processor what general classes of elements the current element belongs to. DITA scopes elements by module type (for example topic type, domain type, or map type) instead of document type. Scoping by module type lets document type developers combine multiple module types in a single document without complicating transformation logic. The sequence of values in the `@class` attribute is important because it tells processors which value

is the most general and which is most specific. This sequence is what enables both specialization aware processing and generalization.

## classification domain elements

The classification domain elements are used to identify the subject matter of content that is referenced in a map. These subjects must be subjects defined in a subject scheme map. These subjects enable an author to classify content in new subject categories that are not bound to existing metadata attributes.

## Classification maps

A classification map is a DITA map in which the classification domain has been made available.

## @collection-type

The @collection-type attribute specifies how the children of a <topicref> element relate to their parent and to each other. This attribute, which is set on the parent element, typically is used by processors to determine how to generate navigation links in the rendered topics. For example, a @collection-type value of "sequence" indicates that children of the specifying <topicref> element represent an ordered sequence of topics; processors might add numbers to the list of child topics or generate next and previous links for online presentation. This attribute is available in topics on the <linklist> and <linkpool> elements, where it has the same behavior. Where the @collection-type attribute is available on elements that cannot directly contain elements (such as <reftable> or <topicref>), the behavior of the attribute is reserved for future use.

## 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.

## concept

The <concept> element is the top-level element for a topic that answers the question "what is?" Concepts provide background information that users must know before they can successfully work with a product or interface. Often, a concept is an extended definition of a major abstraction such as a process or function. It might also have an example or a graphic, but generally the structure of a concept is fairly simple.

## Conditional processing

Conditional processing, also known as profiling, is the filtering or flagging of information based on processing-time criteria. See Conditional processing for more information.

## Conditional processing attributes

The metadata attributes specify properties of the content that can be used to determine how the content is processed. Specialized metadata attributes can be defined to enable specific business-processing needs, such as semantic processing and data mining. Metadata attributes typically are used for the following purposes:

- Filtering content based on the attribute values, for example, to suppress or publish profiled content

- Flagging content based on the attribute values, for example, to highlight specific content on output
- Performing custom processing, for example, to extract business-critical data and store it in a database

Typically `@audience`, `@platform`, `@product`, `@otherprops`, `@props`, `@deliveryTarget`, and specializations of the `@props` attributes are used for filtering; the same attributes plus the `@rev` attribute are used for flagging. The `@status` and `@importance` attributes, including custom attributes specialized from `@base`, are used for application-specific behavior. For example, identifying metadata to aid in search and retrieval.

## Configuration

A document type shell is an XML grammar file that specifies the elements and attributes that are allowed in a DITA document. The document type shell integrates structural modules, domain modules, and constraint modules. In addition, a document type shell specifies whether and how topics can nest. See Configuration for more information.

## conformance

Conformance to the DITA specification allows documents and document types that are used with different processors or different versions of a processor to produce the same or similar results with little or no reimplementation or modification.

## constraint

Constraint modules define additional constraints for vocabulary modules to:

- restrict content models or attribute lists for specific element types
- remove certain extension elements from an integrated domain module
- replace base element types with domain-provided, extension element types

A constraint module can:

- restrict the content model for an element
- restrict available attributes on an element
- restrict elements in a domain
- replace base elements with domain extensions

## constraint module

A set of declarations that imposes additional constraints onto the element or attribute types that are defined in a specific vocabulary module.

## Constraints

Constraint modules define additional constraints for vocabulary modules to restrict content models or attribute lists for specific element types, remove certain extension elements from an integrated domain module, or replace base element types with domain-provided, extension element types. See Constraints for more information.

## Content reference (conref)

The DITA conref attributes provide mechanisms for reusing content. DITA content references support reuse scenarios that are difficult or impossible to implement using other XML-based inclusion



mechanisms like XInclude and entities. Additionally, DITA content references have rules that help ensure that the results of content inclusion remain valid after resolution.

## Content reuse

The DITA `@conref`, `@conkeyref`, `@conrefend`, and `@conaction` attributes provide mechanisms for reusing content within DITA topics or maps. These mechanisms can be used both to pull and push content. See Content reuse for more information.

## Context hooks

User interfaces for software application often are linked to user assistance (such as help systems and tool tips) through context hooks. Context hooks are identifiers that associate a part of the user interface with the location of a help topic. Context hooks can be direct links to URIs, but more often they are indirect links (numeric context identifiers and context strings) that can be processed into external resource files. These external resource and mapping files are then used directly by context-sensitive help systems and other downstream applications.

## Controlled values

Controlled values are keywords that can be used as values for attributes. For example, the `@audience` attribute can take a value that identifies the users that are associated with a particular product. Typical values for a medical-equipment product line might include:

- therapist
- oncologist
- physicist
- radiologist

In a subject scheme map, an information architect can define a list of these values for the `@audience` attribute. Controlled values can be used to classify content for filtering and flagging at build time.

## @copy-to

In most situations, specifies whether a duplicate version of the topic is created when it is transformed. This duplicate version can be either literal or virtual. The value of the `@copy-to` attribute specifies the uniform resource identifier (URI) by which the topic can be referenced by a `@conref` attribute, `<topicref>` element, or `<xref>` element. The duplication is a convenience for output processors that use the URI of the topic to generate the base address of the output. The `@keys` and `@keyref` attributes provide an alternative mechanism; they enable references to topics in specific-use contexts. The `@copy-to` attribute also can be used to specify the name of a new chunk when topics are being chunked. The attribute can also be used to determine the name of the stub topic that is generated from a `<topicref>` element that contains a title but does not specify a target. In both of those cases, no duplicate version of the topic is generated. For information on how the `@copy-to` attribute can be used with the `@chunk` attribute, see Chunking.

## cross-references

Cross-references occur only within the body of a topic and can target any element in this or other topics.

## Darwin Information Typing Architecture (DITA)

The Darwin Information Typing Architecture (DITA) is an XML-based architecture for authoring, producing, and delivering topic-oriented, information-typed content that can be reused, single-sourced, and published in various ways.

### delayed conref resolution

The delayed conref resolution domain provides several elements for use when using DITA in situations that enable delayed or delivery time resolution of conref. The elements allow users to resolve some conref values statically, while delaying others for later resolution.

### @deliveryTarget

The intended delivery target of the content, for example HTML, PDF, or ePub. This attribute is a replacement for the now deprecated @print attribute. The @deliveryTarget attribute is specialized from the @props attribute. It is defined in the deliveryTargetAttDomain, which is integrated into all OASIS-provided document-type shells. If this domain is not integrated into a given document-type shell, the @deliveryTarget attribute is not available.

### descendant scope

A key scope that occurs any level below another key scope.

### @dir

Determines the direction in which the content is rendered.

## DITA addressing

DITA provides two addressing mechanisms. DITA addresses either are direct URI-based addresses, or they are indirect key-based addresses. Within DITA documents, individual elements are addressed by unique identifiers specified on the @id attribute. DITA defines two fragment-identifier syntaxes; one is the full fragment-identifier syntax, and the other is an abbreviated fragment-identifier syntax that can be used when addressing non-topic elements from within the same topic.

### @DITAArchVersion

This attribute identifies the version of the DITA architecture that is used by the XML grammar. The root element of every topic and map MUST declare a @DITAArchVersion attribute. The attribute is declared in a DITA namespace to allow namespace-sensitive tools to detect DITA markup.

## DITA document

An XML document that conforms to the requirements of this specification. A DITA document MUST have as its root element one of the following elements:

- <map> or a specialization of the <map> element
- <topic> or a specialization of the <topic> element
- <dita>, which cannot be specialized, but that allows documents with multiple sibling topics

## DITA document type

A unique set of structural modules, domain modules, and constraint modules that taken together provide the XML element and attribute declarations that define the structure of DITA documents.

## DITA document-type shell

A set of DTD, XSD, or RELAX NG declarations that implement a DITA document type by using the rules and design patterns that are included in the DITA specification. A DITA document-type shell includes and configures one or more structural modules, zero or more domain modules, and zero or more constraint modules. Except for the optional declarations for the `<dita>` element and its attributes, DITA document-type shells do not declare any element or attribute types directly.

## DITA element

An XML element instance whose type is a DITA element type. DITA elements must exhibit a `@class` attribute that has a value that conforms to the rules for specialization hierarchy specifications.

## DITA element type

An element type that is either one of the base element types that are defined by the DITA specification, or a specialization of one of the base element types.

## DITA map attributes

DITA maps have unique attributes that are designed to control the way that relationships are interpreted for different output purposes. In addition, DITA maps share many metadata and linking attributes with DITA topics.

## DITA map elements

A DITA map describes the relationships among a set of DITA topics. The DITA map and map-group domain elements organize topics into hierarchies, groups, and relationships; they also define keys.

## DITA maps

DITA maps are documents that organize topics and other resources into structured collections of information. DITA maps specify hierarchy and the relationships among the topics; they also provide the contexts in which keys are defined and resolved. DITA maps SHOULD have `.ditamap` as the file extension. See DITA maps for more information.

## DITA maps

DITA maps are documents that organize topics and other resources into structured collections of information. DITA maps specify hierarchy and the relationships among the topics; they also provide the contexts in which keys are defined and resolved. DITA maps SHOULD have `.ditamap` as the file extension. See DITA maps for more information.

## DITA topic

In DITA, a topic is the basic unit of authoring and reuse. All DITA topics have the same basic structure: a title and, optionally, a body of content.

## DITA topics

In DITA, a topic is the basic unit of authoring and reuse. All DITA topics have the same basic structure: a title and, optionally, a body of content. See DITA topics for more information.

### <div>

The <div> element enables the arbitrary grouping of content within a topic. The <div> element does not include a title. For content that requires a title, use <section> or <example> or, possibly, <fig>.

## document type declarations (DTDs)

DITA coding practices for DTDs rely heavily on entities to implement specialization and constraints. As such, an understanding of entities is critical when working with DTD document-type shells, vocabulary modules, or constraint modules.

## document-type shell

A document type shell is an XML grammar file that specifies the elements and attributes that are allowed in a DITA document. The document type shell integrates structural modules, domain modules, and constraint modules. In addition, a document type shell specifies whether and how topics can nest.

## domain constraint module

**Comment by scott.hudson**  
Add defn

## domain module

A vocabulary module that defines a set of element types or an attribute type that supports a specific subject or functional area.

## @domains

This attribute identifies the domain modules (and optionally the structural modules) that are used in a map or topic. Each module also declares its module dependencies. The root element of every topic and map MUST declare a @domains attribute.

## effective key definition

The definition for a key within a key space that is used to resolve references to that key. A key might have multiple definitions within a key space, but only one of those definitions is effective.

## effective key definition

The definition for a key within a key space that is used to resolve references to that key. A key might have multiple definitions within a key space, but only one of those definitions is effective.

## element domain module

A domain module that defines one or more element types for use within maps or topics.

## element domain module

A vocabulary module that defines one or more specialized element types that can be integrated with maps or topics. For element domain modules, the module name is typically a name that reflects the subject domain to which the domain applies, such as "highlight" or "software". Domain modules often have an associated short name, such as "hi-d" for the highlighting domain or "sw-d" for the software domain.

## element type pattern

Comment by [scott.hudson](#)  
add defn

## element type specialization

The DITA specialization mechanism provides a standard mechanism for defining that an element type that is derived from an ancestor type. A specialized type inherits the semantics and default processing behavior from its ancestor type. Other processing behavior optionally can be associated with the specialized descendant type.

## enumerationdef element

The `<enumerationdef>` element binds the set of controlled values to an attribute. Valid attribute values are defined in the set of controlled values; invalid attribute values are not defined in the set of controlled values. An enumeration can specify an empty `<subjectdef>` element. In that case, no value is valid for the attribute. An enumeration also can specify an optional default value by using the `<defaultSubject>` element.

## extension element

Within a vocabulary module, an element type that can be extended, replaced, or constrained for use in a DITA document type.

## foreign vocabulary

You can extend DITA to incorporate standard vocabularies for non-textual content, such as MathML and SVG, as markup within DITA documents. This is done by specializing the `<foreign>` or `<unknown>` elements. The `<foreign>` or `<unknown>` elements should not be used to include textual content or metadata in DITA documents, except where such content acts as an example or display, rather than as the primary content of a topic.

## @format

The `@format` attribute identifies the format of the resource being addressed. For example, references to DITA topics are identified with `format="dita"`, whereas references to DITA maps use `format="ditamap"`. References to other types of content should use other values for this attribute. By default, references to non-XML content use the extension of the URI in the `@href` attribute as the effective format.

## generalization

The process by which a specialized element is transformed into a less-specialized ancestor element or a specialized attribute is transformed into a less-specialized ancestor attribute. The original specialization-hierarchy information can be preserved in the generalized instance, allowing the original specialized type to be recreated from the generalized instance.

## Generic topic or Base topic

The element type `<topic>` is the base topic type from which all other topic types are specialized. All topics have the same basic structure.

## highlighting elements

The highlighting elements are used to highlight text with styles (such as bold, italic, and monospace). Never use these elements when a semantically specific element is available. Do not specialize these elements, as they are intended solely for use by authors when no semantically appropriate element is available and a formatting effect is required.

## @href

The `@href` attribute specifies the URI of the resource that is being addressed.

## @id

The `@id` attribute assigns an identifier to DITA elements so that the elements can be referenced. The `@id` attribute is available for most elements. An element must have a valid value for the `@id` attribute before it can be referenced using a fragment identifier. The requirements for the `@id` attribute differ depending on whether it is used on a topic element, a map element, or an element within a topic or map. All values for the `@id` attribute must be XML name tokens.

## Images

Images can be inserted to display photographs, illustrations, screen captures, diagrams, and more. At the phrase level, they can display trademark characters, icons, toolbar buttons, and so forth.

## @importance

The degree of priority of the content. This attribute takes a single value from an enumeration.

## Indirect key-based addressing

DITA also supports indirect links and cross-references in which a DITA map assigns unique names, or keys, to the resources being referenced by the publication. Indirect addressing is done using `<topicref>` elements that specify the `@keys` attribute. Using the `@keyref` attribute, individual links, cross-references, and images then reference resources by their keys instead of their URIs. Links defined using `@keyref` thus allow context-specific linking behavior. That is, the links in a topic or map might resolve to one set of resources in one context, and a different set of resources in another, without the need for any modifications to the link markup. When links are defined using `@keyref`, values for the four linking attributes described above are typically all specified (or given default values) on the key defining element.

## Indirect key-based addressing

DITA keys provide an alternative to direct addressing. The key reference mechanism provides a layer of indirection so that resources (for example, URIs, metadata, or variable text strings) can be defined at the DITA map level instead of locally in each topic.

## Information type

Information typing is the practice of identifying types of topics (such as concept, reference, and task), to clearly distinguish between different types of information. Topics that answer different reader questions (How ...? What is ...?) can be categorized with different information types. The base information types provided by DITA specializations (for example, technical content, machine industry, and learning and training) provide starter sets of information types that many technical and business-related organizations can adopt immediately.

## Information typing

Information typing is the practice of identifying types of topics (such as concept, reference, and task), to clearly distinguish between different types of information. Topics that answer different reader questions (How ...? What is ...?) can be categorized with different information types. The base information types provided by DITA specializations (for example, technical content, machine industry, and learning and training) provide starter sets of information types that many technical and business-related organizations can adopt immediately.

## key definition

A `<topicref>` element that binds one or more key names to zero or more resources.

## key definition

A `<topicref>` element that binds one or more key names to zero or more resources.

## Key definitions

A key definition binds one or more keys to zero or more resources.

## key reference

An attribute that references a key, such as `@keyref` or `@conkeyref`.

## key reference

An attribute that references a key, such as `@keyref` or `@conkeyref`.

## key scope hierarchy

A key scope and all of its descendant scopes.

## key scope

A map or section of a map that defines its key space and serves as the resolution context for its key references.

## key scope

A map or section of a map that defines its key space and serves as the resolution context for its key references.

## key scope

Key scopes enable map authors to specify different sets of key definitions for different map branches. A `<map>` or `<topicref>` element defines a key scope by specifying the `@keyscope` attribute. The `@keyscope` attribute specifies the names of the scope, separated by spaces. A key scope name has the same legal character requirements as for keys.

## Key scopes

All key definitions and key references exist within a key scope. If the `@keyscope` attribute is never specified within the map hierarchy, all keys exist within a single, default key scope.

## Subject definitions

Subject definitions are classifications and sub-classifications that compose a tree. Subject definitions provide semantics that can be used along with taxonomies and ontologies. Along with the classification domain, subject definitions can be used for retrieval and traversal of the content at run time when used with information viewing applications that provide such functionality.

## Key scopes

All key definitions and key references exist within a key scope. If the `@keyscope` attribute is never specified within the map hierarchy, all keys exist within a single, default key scope.

## key space

A list of key definitions that are used to resolve key references.

## key space

A list of key definitions that are used to resolve key references.

## Key spaces

The key space associated with a key scope is used to resolve all key references that occur immediately within that scope. Key references in child scopes are resolved using the key spaces that are associated with those child scopes. A key scope is associated with exactly one key space. That key space contains all key definitions that are located directly within the scope; it might also contain definitions that exist in other scopes.

## key

A name for a resource. See Using keys for addressing for more information.

## key

A name for a resource. See Using keys for addressing for more information.



## <keydef>

Enables authors to define keys. This element is a convenience element; it is a specialization of `<topicref>` that sets the default value of the `@processing-role` attribute to "resource-only." Setting the `@processing-role` attribute to `resource-only` ensures that the resource referenced by the key definition is not directly included in the navigation that is defined by the map.

## @keys

Specifies one or more key names.

## @keyscope

Defines a new scope for key definition and resolution, and gives the scope one or more names. For more information about key scopes, see Indirect key-based addressing.

## link

The `<link>` element defines a relationship to another topic or non-DITA resource. When displayed, links are typically sorted based on their attributes, which define the type or role of the link's target in relation to the current topic.

## @linking

By default, the relationships between the topics that are referenced in a map are reciprocal:

- Child topics link to parent topics and vice versa.
- Next and previous topics in a sequence link to each other.
- Topics in a family link to their sibling topics.
- Topics referenced in the table cells of the same row in a relationship table link to each other.

A topic referenced within a table cell does not (by default) link to other topics referenced in the same table cell. This behavior can be modified by using the `@linking` attribute, which enables an author or information architect to specify how a topic should participate in a relationship. The following values are valid:

### - `linking="none"`

Specifies that the topic does not exist in the map for the purposes of calculating links.

### - `linking="sourceonly"`

Specifies that the topic will link to its related topics but not vice versa.

### - `linking="targetonly"`

Specifies that the related topics will link to it but not vice versa.

### - `linking="normal"`

Default value. It specifies that linking will be reciprocal (the topic will link to related topics, and they will link back to it). Authors also can create links directly in a topic by using the `<xref>` or `<link>` elements, but in most cases map-based linking is preferable, because links in topics create dependencies between topics that can hinder reuse.

Note that while the relationships between the topics that are referenced in a map are reciprocal, the relationships merely imply reciprocal links in generated output that includes links. The rendered navigation links are a function of the presentation style that is determined by the processor.

## links

Links only represent topic-to-topic connections, or connections to non-DITA resources. Links occur after the body of a topic, in the related-links element. Links can also be managed indirectly using DITA maps, which provide a more efficient way to manage links. Using maps to define links; avoids embedded pointers in each topic.

## localization

DITA has features that facilitate preparing content for translation and working with multilingual content, including the `@xml:lang` attribute, the `@dir` attribute, and the `@translate` attribute. In addition, the `<sort-as>` and `<index-sort-as>` elements support sorting for languages in which the correct sorting of an element requires text that is different from the base content of the element.

## @locktitle

If `@locktitle` is set to yes, the `<navtitle>` element or `@navtitle` attribute is used if it is present. Otherwise, the `<navtitle>` element or `@navtitle` attribute is ignored and the navigation title is retrieved from the referenced file. Note: The `@navtitle` attribute is deprecated in favor of the `<navtitle>` element. When both a `<navtitle>` element and a `@navtitle` attribute are specified, the `<navtitle>` element should be used.

## map branch

A `<topicref>` element or a specialization of `<topicref>`, along with any child elements and all resources that are referenced by the original element or its children.

## map group elements

The map group domain elements define, group, or reference content. Many of the map group elements are convenience elements, in other words, they make it easier for an author to make use of existing functions.

## map instance

An occurrence of a map type in a DITA document.

## map type

A map or a specialization of map that defines a set of relationships among topic instances.

## <map>

The `<map>` element is the root element of the DITA map.

## map

The `<map>` element describes the relationships among a set of resources, such as DITA topics. Maps consist of references to topics, maps, and other resources organized into hierarchies, groups, and tables. Maps express these relationships in a single common format that can be used for different outputs.

## <mapref>

Enables authors to reference an entire DITA map, including hierarchy and relationship tables. This element is a convenience element; it is a specialization of <topicref> that sets the default value of the @format attribute to ditamap. The <mapref> element represents a reference from a parent map to a subordinate map.

## Metadata elements

The metadata elements, many of which map to Dublin core metadata, are available in topics and DITA maps. This design enables authors and information architects to use identical metadata markup in both topics and maps. The <metadata> element is a wrapper element that contains many of the metadata elements. In topics, the <metadata> element is available in the <prolog> element. In maps, the <metadata> element is available in the <topicmeta> element. In DITA maps, the metadata elements also are available directly in the <topicmeta> element. Collections of metadata can be shared between DITA maps and topics by using the conref or keyref mechanism. In general, specifying metadata in a <topicmeta> element is equivalent to specifying it in the <prolog> element of a referenced topic. The value of specifying the metadata at the map level is that the topic then can be reused in other maps where different metadata might apply. Many items in the <topicmeta> element also cascade to nested <topicref> elements within the map.

**Note** Not all metadata elements are available in the <metadata> element. However, they are available in either the topic <prolog> element or the map <topicmeta> element.

## metadata elements

Metadata elements include information that is located within the <topicmeta> element (in maps) or <prolog> element (in topics), and indexing elements that can be placed in other locations within topic content.

## modularization

Modularization is at the core of DITA design and implementation. It enables reuse and extension of the DITA specialization hierarchy. The DITA XML grammar files are a set of module files that declare the markup and entities that are required for each specialization. The document-type shell then integrates the modules that are needed for a particular authoring and publishing context.

## module inclusion

## Multimedia

The <object> element enables authors to include multimedia, such as diagrams that can be rotated and expanded. The <foreign> element enables authors to include media within topic content, for example, SVG graphics, MathML equations, and so on.

## Navigation

DITA includes markup that processors can use to generate reader navigation to or across DITA topics. Such navigation behaviors include table of contents (TOCs), relationship tables, and indexes.

## <navref>

The `<navref>` element represents a pointer to another map, which is preserved as a transcluding link in the result deliverable rather than resolved when the deliverable is produced. Output formats that support such linking can integrate the referenced resource when displaying the referencing map to an end user.

## @navtitle

Specifies a navigation title. This is a shorter version of the title that is used in the navigation only. By default, the `@navtitle` attribute is ignored; it serves only to help the DITA map author keep track of the title of the topic.

**Note** The `@navtitle` attribute is deprecated in favor of the `<navtitle>` element. When both a `<navtitle>` element and a `@navtitle` attribute are specified, the `<navtitle>` element should be used.

## Nested topics

Topics can be defined inside other topics. However, nesting requires special care because it can result in complex documents that are less usable and less reusable. Nesting might be appropriate for information that is first converted from desktop publishing or word processing files or for topics that are unusable independent from their parent or sibling topics. The rules for topic nesting can be configured in a document-type shell. For example, the standard DITA configuration for concept topics only allows nested concept topics. However, local configuration of the concept topic type could allow other topic types to nest or disallow topic nesting entirely. In addition, the `@chunk` attribute enables topics to be equally reusable regardless of whether they are separate or nested. The standard DITA configuration for ditabase document-type documents allows unrestricted topic nesting and can be used for holding sets of otherwise unrelated topics that hold reusable content. It can also be used to convert DITA topics from non-DITA legacy source without first determining how individual topics should be organized into separate XML documents.

## nesting

### Non-normative information

Non-normative information includes descriptions that provide background, examples, notes, and other useful information that are not formal requirements or rules that must be followed.

### Normative information

Normative information is the formal portion of the specification that describes the rules and requirements that make up the DITA standard and which must be followed.

## @otherprops

Other properties that do not require semantic identification.

## @outputclass

Provides a label on one or more element instances, typically to specify a role or other semantic distinction. As the `@outputclass` attribute does not provide a formal type declaration or the structural consistency of specialization, it should be used sparingly, usually only as a temporary measure while a

specialization is developed. For example, `<uicontrol>` elements that define button labels could be distinguished by adding an `@outputclass` attribute:

```
<uicontrol outputclass="button">Cancel</uicontrol>
```

The value of the `@outputclass` attribute can be used to trigger XSLT or CSS rules, while providing a mapping to be used for future migration to a more specialized set of user interface elements.

## parameter entity

## parent scope

A key scope that occurs one level above another key scope.

## pattern

## peer map

A DITA map that is referenced with a `@scope` attribute that evaluates as "peer". The value of the scope attribute might be explicitly set, be defaulted, or cascade from another element.

## Phrases and keywords

Phrase level elements can contain markup to label parts of a paragraph or parts of a sentence as having special semantic meaning or presentation characteristics, such as `<uicontrol>` or `<b>`. Phrases can usually contain other phrases and keywords as well as text. Keywords can only contain text.

## @platform

The platform on which the product is deployed.

## @print

Specifies whether the topic should be included in printed output. Note: Beginning with DITA 1.3, the `@print` attribute is deprecated. It is replaced with a conditional processing attribute: `@deliveryTarget`. See `@deliveryTarget` for more details.

## @processing-role

Specifies whether the topic or map referenced should be processed normally or treated as a resource that is only included to resolve key or content references.

### **processing-role="normal"**

The topic is a readable part of the information set. It is included in navigation and search results. This is the default value for the `<topicref>` element.

### **processing-role="resource-only"**

The topic should be used only as a resource for processing. It is not included in navigation or search results, nor is it rendered as a topic. This is the default value for the `<keydef>` element.

- <> - " If the `@processing-role` attribute is not specified locally, the value cascades from the closest element in the containment hierarchy.

## @product

The product that is the subject of the discussion.

## profiling

Conditional processing, also known as profiling, is the filtering or flagging of information based on processing-time criteria. DITA defines attributes that can be used to enable filtering and flagging individual elements. The @audience, @deliveryTarget, @otherprops, @platform, and @props attributes (along with specializations of @props) allow conditions to be assigned to elements so that the elements can be included, excluded, or flagged during processing. The @rev flagging attribute allows values to be assigned to elements so that special formatting can be applied to those elements during processing. A conditional-processing profile specifies which elements to include, exclude, or flag. DITA defines a document type called DITAVAL for creating conditional-processing profiles.

## <prolog>

The prolog is the container for topic metadata, such as change history, audience, product, and so on.

## @props

A generic conditional processing attribute that can be specialized to create new semantic conditional-processing attributes.

## referenced element

An element that is referenced by another DITA element. See also referencing element.

## <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.

## 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.

## referencing element

An element that references another DITA element by specifying an addressing attribute. See also referenced element and addressing attribute.

## Related links

Related links connect to other topics. When an author creates a link as part of a topic, the topic becomes dependent on the other topic being available. To reduce dependencies between topics and thereby

increase the reusability of each topic, authors can use DITA maps to define and manage links between topics, instead of embedding links directly in each related topic.

## RelaxNG

[INSERT DEF] RELAX NG modules are self-integrating, which means that they automatically add to the content models and attribute sets they extend. This means that information architects do not have much work to do when assembling vocabulary modules and constraints into document type shells.

### <reltable>

Relationship tables are defined with the `<reltable>` element. Relationship tables can be used to define relationships among DITA topics or among DITA topics and non-DITA resources. In a relationship table, the columns define common attributes, metadata, or information types (for example, task or troubleshooting) for the resources that are referenced in that column. The rows define relationships between the resources in different cells of the same row. The `<relrow>`, `<relcell>`, `<relheader>`, and `<relcolspec>` elements are used to define the components of the relationship table. Relationships defined in the relationship table also can be further refined by using the `@collection-type` attribute.

### resource

For the purposes of keys and key resolution, a resource includes one of the following items:

- An object addressed by URI
- Metadata specified on a resource, such as a `@scope` or `@format` attribute
- Text or metadata located within a `<topicmeta>` element

### resource

For the purposes of keys and key resolution, one of the following - An object addressed by URI - Metadata specified on a resource, such as a `@scope` or `@format` attribute - Text or metadata located within a `<topicmeta>` element

### <resourceid> element

Context hook information specified in the `<resourceid>` element in the DITA map or in a DITA topic enables processors to generate the header, map, alias, and other types of support files that are required to integrate the user assistance with the application.

### @rev

The revision or draft number of the current document. (This attribute is used only for flagging.)

### root element

### root map

The DITA map that is provided as input for a processor.

## <schemeref>

The `<schemeref>` element provides a mechanism for extending a subject scheme. This makes it possible to add new relationships to existing subjects and extend enumerations of controlled values. The `<schemeref>` element provides a reference to another subject scheme map. Typically, the referenced subject-scheme map defines a base set of controlled values that are extended by the current subject-scheme map. The values in the referenced subject-scheme map are merged with the values in the current subject-scheme map; the result is equivalent to specifying all of the values in a single subject scheme map.

## @scope

The `@scope` attribute describes the closeness of the relationship between the current document and the target resource. Resources in the same information unit are considered "local"; resources in the same system as the referencing content but not part of the same information unit are considered "peer"; and resources outside the system, such as Web pages, are considered "external".

## scope-qualified key name

A scope-qualified key name is a key name, prepended by one or more key scope names and separated by periods. For example, to reference a key "keyName" defined in a child scope named "keyScope", specify `keyref="keyScope.keyName"`.

## @search

Specifies whether the topic should be included in search indexes.

## <section>

The `<section>` element represents an organizational division in a topic. Sections are used to organize subsets of information that are directly related to the topic. Multiple `<section>` elements within a single topic do not represent a hierarchy, but rather peer divisions of that topic. Sections cannot be nested. A `<section>` can have an optional `<title>`.

## <sectiondiv>

The `<sectiondiv>` element enables the arbitrary grouping of content within a section for the purpose of content reuse. The `<sectiondiv>` element does not include a title. Content that requires a title should use `<section>` or `<example>`.

## Sections and examples

The body of a topic might contain divisions, such as sections and examples. They might contain block-level elements like titles and paragraphs and phrase-level elements like API names or text. It is recommended that sections have titles, whether they are entered directly into the `<title>` element or rendered using a fixed or default title. Either body divisions or untitled sections or examples can be used to delimit arbitrary structures within a topic body. However, body divisions can nest, but sections and examples cannot contain sections.



## Short description or abstract

A short description of the topic or a longer abstract with an embedded short description. The short description might be used both in topic content (as the first paragraph), in generated summaries that include the topic, and in links to the topic. Alternatively, the abstract lets you create more complex introductory content and uses an embedded short description element to define the part of the abstract that is suitable for summaries and link previews. While short descriptions aren't required, they can make a dramatic difference to the usability of an information set and should generally be provided for all topics.

### <shortdesc>

The <shortdesc> element provides a short description of the topic. The short description, which represents the purpose or theme of the topic, is also intended to be used as a link preview and for search results. The element can occur both in topics and maps. The short description should be a single, concise paragraph containing one or two sentences of no more than 50 words.

## sibling scope

A key scope that shares a common parent with another key scope.

## simpletable attribute group

The simpletable attribute group includes several attributes that are defined on the <simpletable> element and <simpletable> specializations.

### <simpletable>

The <simpletable> element is used for tables that are regular in structure and do not need a caption.

## sorting

Processors can be configured to sort elements. Typical processing includes sorting glossary entries, lists of parameters or reference entries in custom navigation structures, and tables based on the contents of cells in specific columns or rows. The <sort-as> element can be used to specify an effective sort phrase when the base sort phrase is not appropriate for sorting. For index terms, the <index-sort-as> element can be used to specify the effective sort phrase for an index entry. When a <sort-as> element is specified, processors that sort the containing element MUST construct the effective sort phrase by prepending the content of the <sort-as> element to the base sort phrase. This ensures that two items with the same <sort-as> element but different base sort phrases can sort in the appropriate order.

## specialization hierarchy

The sequence of element or attribute types, from the most general to most specialized, from which a given element or attribute type is specialized. The specialization hierarchy for a DITA element is formally declared through its @class attribute.

## specialization

(1) The act of defining new element or attribute types as a semantic refinement of existing element or attribute types (2) An element or attribute type that is a specialization of a base type (3) A process by which a generalized element is transformed into one of its more specialized element types or a generalized attribute is transformed into a more specialized attribute

## Specialization

The specialization feature of DITA enables the creation of new element types and attributes that are explicitly and formally derived from existing types. This facilitates interchange of conforming DITA content and ensures a minimum level of common processing for all DITA content. It also allows specialization-aware processors to add specialization-specific processing to existing base processing. See Specialization for more information.

## specialization

The specialization feature of DITA enables the creation of new element types and attributes that are explicitly and formally derived from existing types. This facilitates interchange of conforming DITA content and ensures a minimum level of common processing for all DITA content. It also allows specialization-aware processors to add specialization-specific processing to existing base processing. Specialization modules enable information architects to create new element types and attributes. These new element types and attributes are derived from existing element types and attributes.

## @status

The current state of the content. This attribute takes a single value from an enumeration.

## structural module

A vocabulary module that defines a top-level map type or topic type.

## structural type instance

An occurrence of a topic type or a map type in a DITA document.

## structural type

A topic type or map type.

## structural vocabulary module

A vocabulary module that defines a top-level map or topic type. Structural modules also can define specializations of, or reuse elements from, domain or other structural modules. When this happens, the structural module becomes dependent.

## subject scheme elements

Subject scheme maps are used to define sets of controlled values and taxonomic subjects; bind controlled values to attributes; and specify relationships among taxonomic subjects.

## Subject scheme maps

Subject scheme maps can be used to define controlled values and subject definitions. The controlled values can be bound to attributes, and 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.

## Subject scheme maps

Subject scheme maps use key definitions to define collections of controlled values and subject definitions.

### <subjectdef>

Each controlled value is defined using a <subjectdef> element, which is a specialization of the <topicref> element. The <subjectdef> element is used to define both a subject category and a list of controlled values. The parent <subjectdef> element defines the category, and the children <subjectdef> elements define the controlled values.

### submap

A DITA map that is referenced with a @scope attribute that evaluates as "local". The value of the scope attribute might be explicitly set, be defaulted, or cascade from another element.

### <table>

The <table> element organizes arbitrarily complex relationships of tabular information. This standard table markup allows column or row spanning and table captions or descriptions. An optional title allowed inside the <table> element provides a caption to describe the table. The DITA table is based on the OASIS Exchange Table Model, augmented with DITA attributes that enable it for accessibility, specialization, conref, and other DITA processing. In addition, the table includes a <desc> element, which enables table description that is parallel with figure description.

### <task>

The <task> element is the top-level element for a task topic. Tasks are the main building blocks for task-oriented user assistance. They generally provide step-by-step instructions that enable you to perform a task. A task answers the question of "how to?" by telling the user precisely what to do and the order in which to do it. Tasks have the same high-level structure as other topics, with a title, short description, and body.

## text entity

### Title

The title contains the subject of the topic.

### @toc

Specifies whether topics are excluded from navigation output, such as a website map or an online table of contents. By default, <topicref> hierarchies are included in navigation output; relationship tables are excluded.

## Topic body

The topic body contains most content except for content contained in the title or the short description or abstract. The topic body can be constrained to remove specific elements from the content model; it also can be specialized to add additional specialized elements to the content model. The topic body can be generic while the topic title and prolog are specialized.

## Topic element

The topic element holds the required `@id` attribute and contains all other elements.

## topic instance

An occurrence of a topic type in a DITA document.

## Topic-oriented writing

Topic-oriented writing is a disciplined approach to writing that emphasizes modularity and reuse of concise units of information: topics. Well-designed DITA topics can be reused in many contexts, as long as writers are careful to avoid unnecessary transitional text.

## @type

The `@type` attribute is used on cross-references to describe the target of the reference. Most commonly, the `@type` attribute names the element type being referenced when `format="dita"`.

## <topicgroup>

The `<topicgroup>` element defines a group or collection outside of a hierarchy or relationship table, and is a convenience element that is equivalent to a `<topicref>` element without an `@href` attribute or navigation title. Groups can be combined with hierarchies and relationship tables, for example, by including a `<topicgroup>` element within a set of siblings in a hierarchy or within a table cell. The `<topicref>` elements so grouped can then share inherited attributes and linking relationships with no effect on the navigation or table of contents.

## <topichead>

The `<topichead>` element provides a navigation title, and is a convenience element that is equivalent to a `<topicref>` element with a navigation title but no associated resource.

## <topicmeta>

Most map-level elements, including the map itself, can contain metadata inside the `<topicmeta>` element. Metadata typically is applied to an element and its descendants.

## <topicref>

The `<topicref>` elements are the basic elements of a map. A `<topicref>` element can reference a DITA topic, a DITA map, or a non-DITA resource. A `<topicref>` element also can have a title, short description, and the same kind of prolog-level metadata that is available in topics. The `<topicref>` elements can be nested to create a hierarchy, which can be used to define a table of contents (TOC) for print output, online navigation, and parent/child links. Hierarchies can be annotated using the `@collection-type` attribute to define a particular type of relationship, such as a set of choices, a sequence, or a family. These collection types can affect link generation, and they might be interpreted differently for different outputs.

## <topicset>

Enables authors to define a branch of navigation in a DITA map so that it can be referenced from another DITA map.

## <topicsetref>

Enables authors to reference a navigation branch that is defined in the current map or in another DITA map.

## @translate

Determines whether the element requires translation. A default value can often be inferred from the element type. For example, <apiname> might be untranslated by default, whereas <p> might be translated by default.

## <troubleshooting>

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

Troubleshooting topics document corrective action such as troubleshooting or alarm clearing.

## topic type

A topic or a specialization of topic that defines a complete unit of content.

## URI-based addressing

### URI references

Content reference and link relationships can be established from DITA elements by using URI references. DITA uses URI references in @href, @conref, and other attributes for all direct addressing of resources. URI references address resources and (in some cases) subcomponents of those resources. In this context, a resource is a DITA document (map, topic, or DITA base document) or a non-DITA resource (for example, an image, a Web page, or a PDF document).

### URIs and fragment identifiers

The fragment identifier is the part of the URI that starts with a number sign (#), for example, #topicid/elementid. URI references also can include a query component that is introduced with a question mark (?).

## <ux-window>

In some help systems, a topic might need to be displayed in a specifically sized or featured window. For example, a help topic might need to be displayed immediately near the user interface control that it supports in a window of a specific size that always remains on top, regardless of the focus within the

operating system. Windowing metadata can be defined in the DITA map within the `<ux-window>` element. The `<ux-window>` element provides the `@top`, `@left`, `@height`, `@width`, `@on-top`, `@features`, `@relative`, and `@full-screen` attributes.

## <ux-window>

The `<ux-window>` element enables authors to define windowing information for the display of output topics that are appropriate to the delivery platform. Window management is important in user assistance and help system outputs, as well as for other hypertext and electronic delivery modes.

## vocabulary module

A set of element or attribute declarations.

## vocabulary module

A DITA element type or attribute is declared in exactly one vocabulary module. The following terminology is used to refer to DITA vocabulary modules:

### structural module

A vocabulary module that defines a top-level map or topic type. Structural modules also can define specializations of, or reuse elements from, domain or other structural modules. When used in this way, the structural module becomes dependent.

### element domain module

A vocabulary module that defines one or more specialized element types that can be integrated with maps or topics.

### attribute domain module

A vocabulary module that defines exactly one specialization of either the `@base` or `@props` attribute.

## XML grammar

### @xml:lang

Identifies the language of the content, using the standard language and country codes. For instance, French Canadian is identified by the value `fr-CA`. The `@xml:lang` attribute asserts that all content and attribute values within the element bearing the attribute are in the specified language, except for contained elements that declare a different language.

## XML Schema declarations (XSDs)

**Comment by scott.hudson**  
{INSERT DEF}

DITA coding practices for XML Schema rely on the XSD redefine facility to implement specializations or constraints. However, limitations in the redefine facility can present problems for some DITA modules implemented in XML Schema.

## <xref>

Use the `<xref>` element to provide an inline cross-reference. The element is commonly used to link to a different location within the current topic, a different topic, a specific location in another topic, or an external resource. The target of the cross-reference is specified using the `@href` or `@keyref` attributes.