Review U: Overview of DITA

Table of contents

1 Overview of 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 and single-sourced in a variety of ways. While DITA historically has been driven by the requirements of large-scale technical documentation authoring, management, and delivery, it is a standard that is applicable to any kind of publication or information that might be presented to readers, including interactive training and educational materials, standards, reports, business documents, trade books, travel and nature guides, and more.

Comment by Kristen James Eberlein on 01 August 20223

Do we really want or need the 2nd sentence? Can we shorten it and remove the reference to technical documentation? It just strikes me as dated, and I suspect that it was added for DITA 1.2 (2010).

Leroy: I think we can remove examples after "including" and end sentence after "any kind of publication"

Dawn: I personally don't think we need the sentence at all.

Disposition: Unassigned

Comment by Christina Rothwell
August 4, 2023 I agree. DITA can be used for any type of publication. I like the idea of shortening this and providing the main concept in that different types of publications can be created from a single source. In our notes, we mentioned about removing "trade books, travel and nature guides, and more."
Disposition: Unassigned

DITA is designed for creating new document types and describing new information domains based on existing types and domains. The process for creating new types and domains is called specialization. Specialization enables the creation of specific, targeted XML grammars that can still use tools and design rules that were developed for more general types and domains; this is similar to how classes in an object-oriented system can inherit the methods of ancestor classes.

Comment by dstevens73
This seems to me to take a giant step from the basic concept of what is DITA. We haven't defined document types and information domains and we're saying you can create new ones. I think we should get through the majority of this chapter – what are topics and maps before we say you can design and specialize new ones.
Disposition: Unassigned

Because DITA topics are conforming XML documents, they can be readily viewed, edited, and validated using standard XML tools, although realizing the full potential of DITA requires using DITA-aware tools.

Comment by Kristen J Eberlein on 03 June 2019

This section of the spec now contains material about topics, maps, and metadata that was previously in the "DITA markup" section."

We need to carefully consider what of this content is appropriate. Some of it – information about map elements and attributes, metadata – is duplicated elsewhere. If we think it is useful to have a high-level overview here, we should mark it as non-normative – and point users to the normative coverage of the topic.

In a parallel move, I think we'll need to move coverage of critical DITA attributes into a more prominent place in the spec.

Dawn:

Disposition: Unassigned

1.1 Basic concepts

DITA has been designed to satisfy requirements for information typing, semantic markup, modularity, reuse, interchange, and production of different deliverable forms from a single source. These topics provide an overview of the key DITA features and facilities that serve to satisfy these requirements.

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. Topics can be generic or more specialized; specialized topics represent more specific information types or semantic roles, for example, `<concept>`, `<task>`, or `<reference>`

Comment by Stan Doherty
Need terminating period before "See".
Disposition: Unassigned

See DITA topics (7) 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. See DITA maps (13) for more information.

Comment by Stan Doherty
Suggest removing the semicolon. Separate thoughts.
Disposition: Unassigned

Comment by Nancy Harrison
we mention keys in this section before we've introduced the concept of what a key is. I think we might want to say 'in which indirect references - keys - can be defined and resolved.' Then we should s-ref DITA addressing as well as maps
Disposition: Unassigned

Information typing

Comment by Christina Rothwell
Christina August 1, 2023: Should we keep the information typing topic?
Disposition: Unassigned

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 do I? 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 can be adopted immediately by many technical and business-related organizations. See Information typing (10) for more information.

Comment by Nancy Harrison
again, we mention specializations here before we've introduced the concept of what a key is. I'd replace the 3rd sentence with something like 'The base information type is a topic, which can be specialized to create more specific types of information; the standard, as delivered, includes some specialized topics, such as task, concept, and reference.' Then x-ref specialization (below) as well as information typing
Disposition: Unassigned

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. See DITA addressing for more information.

Comment by Stan Doherty
In the first sentence, suggest "DITA addresses are either . . . ".
Disposition: Unassigned

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

Conditional processing

Conditional processing is the filtering or flagging of information based on processing-time criteria. See Conditional processing for more information.

Configuration

Comment by Christina Rothwell
Christina Rothwell August 4, 2023 Do we need the hyperlinks in the Configuration, Specialization, and Constraints sections?
Disposition: Unassigned

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 element-configuration modules. In addition, a document-type shell specifies whether and how topics can nest. See Document-type configuration for more information.

Specialization

The specialization feature of DITA allows for 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.

Comment by Stan Doherty
Suggest "This facilitates the interchange . . . "
Disposition: Unassigned

Constraints

Constraint modules 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.

Comment by Eric Sirois

Kris Eberlein, 01 August 2023

Good point. If we keep this topic, we'll need to update it. Does this topic add value?

Would it be better if it did not link out to other sections in the spec? If it did not have those inline links – or if we filtered them out – this "Overview of DITA" submap might be something we could publish as a DITA 101-type publication.

Disposition: Unassigned

Comment by Nancy Harrison
I'd replace this with a section on element configuration, since in 2.0 we're expanding the constraints mechanism to enable both constraints and extensions
Disposition: Unassigned

1.2 Producing different deliverables from a single source

DITA is designed to enable the production of multiple deliverable formats from a single set of DITA content. This means that many rendition details are specified neither in the DITA specification nor in the DITA content; the rendition details are defined and controlled by the processors.

Comment by Stan Doherty
Suggest removing the previous semicolon.
Disposition: Unassigned

Comment by Leroy Steinbacher
Is this about publishing or reuse? Seems to cover both. Some of the definitions below may also be redundant if mentioned somewhere else.
Disposition: Unassigned

Comment by Christina Rothwell
August 4, 2023. I agree. Maybe this topic needs to focus on the process with multichannel publishing.
Disposition: Unassigned

Like many XML-based applications for human-readable documentation, DITA supports the separation of content from presentation. This is necessary when content is used in different contexts, since authors cannot predict how or where the material that they author will be used. The following features and mechanisms enable users to produce different deliverable formats from a single source:

DITA maps

Different DITA maps can be optimized for different delivery formats. For example, you might have a book map for printed output and another DITA map to generate online help; each map uses the same content set.

Specialization

The DITA specialization facility enables users to create XML elements that can provide appropriate rendition distinctions. Because the use of specializations does not impede interchange or interoperability, DITA users can safely create the specializations that are demanded by their local delivery and rendition requirements, with a minimum of additional impact on the systems and business processes that depend on or use the content. While general XML practices suggest that element types should be semantic, specialization can be used to define element types that are purely presentational in nature. The highlighting domain is an example of such a specialization.

Conditional processing
Conditional processing makes it possible to have a DITA topic or map that contains delivery-specific content.

Content referencing
The conref mechanism makes it possible to construct delivery-specific maps or topics from a combination of generic components and delivery-context-specific components.

Key referencing
The keyref mechanism makes it possible to have key words be displayed differently in different deliverables. It also allows a single link to resolve to different targets in different deliverables.

@outputclass attribute
The `@outputclass` attribute provides a mechanism whereby authors can indicate specific rendition intent where necessary. Note that the DITA specification does not define any values for the `@outputclass` attribute; the use of the `@outputclass` attribute is processor specific.

While DITA is independent of any particular delivery format, it is a standard that supports the creation of human-readable content. As such, it defines some fundamental document components including paragraphs, lists, and tables. When there is a reasonable expectation that such basic document components be rendered consistently, the DITA specification defines default or suggested renderings.

Comment by Stan Doherty
Relative to the first sentence, dare we claim (justifiably) that DITA content also machine-readable.
Disposition: Unassigned

1.3 DITA topics

DITA topics are the basic units of DITA content and the basic units of reuse. Each topic contains a single subject.

1.3.1 The topic as the basic unit of information

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. Topics can be generic or more specialized; specialized topics represent more specific information types or semantic roles, for example, `<concept>`, `<task>`, or `<reference>`

DITA topics consist of content units that can be as generic as sets of paragraphs and unordered lists or as specific as sets of instructional steps in a procedure or cautions to be considered before a procedure is performed. Content units in DITA are expressed using XML elements and can be conditionally processed using metadata attributes.

Comment by Leroy Steinbacher on 01 Aug 2023
Remove "Classically"
Disposition: Unassigned

Classically, a DITA topic is a titled unit of information that can be understood in isolation and used in multiple contexts. It is short enough to address a single subject or answer a single question but long enough to make sense on its own and be authored as a self-contained unit. However, DITA topics also can be less self-contained units of information, such as topics that contain only titles and short descriptions and serve primarily to organize subtopics or links or topics that are designed to be nested for the purposes of information management, authoring convenience, or interchange.

DITA topics are used by reference from DITA maps. DITA maps enable topics to be organized in a hierarchy for publication. Large units of content, such as complex reference documents or book chapters, are created by nesting topic references in a DITA map. The same set of DITA topics can be used in any number of maps.

DITA topics also can be used and published individually; for example, one can represent an entire deliverable as a single DITA document that consists of a root topic and nested topics. This strategy can accommodate the migration of legacy content that is not topic-oriented; it also can accommodate information that is not meaningful outside the context of a parent topic. However, the power of DITA is most fully realized by storing each DITA topic in a separate XML document and using DITA maps to organize how topics are combined for delivery. This enables a clear separation between how topics are authored and stored and how topics are organized for delivery.

1.3.2 The benefits of a topic-based architecture

Topics enable the development of usable and reusable content.

Comment by Christina Rothwell
August 4, 2023

We are now realizing not everyone goes to DITA because of reuse. Some writers are benefiting from DITA and they are not doing reuse. They like the consistency of DITA. We don't want to imply that you are not really doing DITA if you are not reusing.

The benefits of DITA go beyond reuse.

If we are going to keep this topic, this topic would need to talk about other benefits of DITA and if so, move this topic out of the DITA topic section.

Or rewrite the heading to reflect more of the content.

Benefits of DITA:

* Reuse

* Consistency of information

* Smaller chunks

* Being able to send a small section of the document for a translation update.

Disposition: Unassigned

While DITA does not require the use of any particular writing practice, the DITA architecture is designed to support authoring, managing, and processing of content that is designed to be reused. Although DITA provides significant value even when reuse is not a primary requirement, the full value of DITA is realized when content is authored with reuse in mind. To develop topic-based information means creating units of standalone information that are meaningful with little or no surrounding context.

By organizing content into topics that are written to be reusable, authors can achieve several goals:

- Content is readable when accessed from an index or search, not just when read in sequence as part of an extended narrative. Since most readers do not read technical and business-related information from beginning to end, topic-oriented information design ensures that each unit of information can be read independently.

- Content can be organized differently for online and print delivery. Authors can create task flows and concept hierarchies for online delivery and create a print-oriented hierarchy to support a narrative content flow.
- Content can be reused in different collections. Since a topic is written to support random access (as by search), it should be understandable when included as part of various product deliverables. Topics permit authors to refactor information as needed, including only the topics that apply to each unique scenario.
- Content is more manageable in topic form whether managed as individual files in a traditional file system or as objects in a content management system.
- Content authored in topics can be translated and updated more efficiently and less expensively than information authored in larger or more sequential units.
- Content authored in topics can be filtered more efficiently, encouraging the assembly and deployment of information subsets from shared information repositories.

Topics written for reuse should be small enough to provide opportunities for reuse but large enough to be coherently authored and read. When each topic is written to address a single subject, authors can organize a set of topics logically and achieve an acceptable narrative content flow.

1.3.3 Disciplined, topic-oriented writing

Comment by Christina Rothwell
Christina Rothwell August 1, 2023: This topic is kind of preachy. We do not want writing best practices in the specification because a lot of other resources are out there for writing best practices for DITA and topic writing. Also each company will have their own best practices.
Disposition: Unassigned

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.

Conciseness and appropriateness

Readers who are trying to learn or do something quickly appreciate information that is written in a structure that is easy to follow and contains only the information needed to complete that task or grasp a fact. Recipes, encyclopedia entries, car repair procedures; all serve up a uniquely focused unit of information. The topic contains everything required by the reader.

Locational independence

A well-designed topic is reusable in other contexts to the extent that it is context free, meaning that it can be inserted into a new document without revision of its content. A context-free topic avoids transitional text. Phrases like "As we considered earlier" or "Now that you have completed the initial step" make little sense if a topic is reused in a new context in which the relationships are different or no longer exist. A well-designed topic reads appropriately in any new context because the text does not refer the reader outside the topic.

Navigational independence

Most print publications or web pages are a mixture of content and navigation. Internal links lead a reader through a sequence of choices as he or she navigates through a website. DITA supports the separation of navigation from content by assembling independent topics into DITA maps. Nonetheless, writers might want to provide links within a topic to additional topics or external resources. DITA does not prohibit such linking within individual topics. The DITA relationship table enables links between topics and to external content. Since it is defined in the DITA map, it is managed independently of the topic content.

Links in the content are best used for cross-references within a topic. Links from within a topic to additional topics or external resources are best avoided because they limit reuse of the topic. To link from a term or keyword to its definition, use the DITA keyref facility to avoid creating topic-to-topic dependencies that are difficult to maintain. See Key-based addressing

Comment by Leroy Steinbacher on 8/1/23
Do we want to go into the details of best practices? I think we could keep this general outline, but remove the specific "preachy" examples
Disposition: Unassigned

1.3.4 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 do I? 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 can be adopted immediately by many technical and business-related organizations.

Comment by Nancy Harrison
this duplicate s the content from the 'Basic concepts' section. I'd guess we want just part of the first sentence 'Information typing is the practice of identifying types of topics to clearly distinguish between different types of information. ' in the basic concepts section, and leave what's here as is.
Disposition: Unassigned

Information typing has a long history of use in the technical documentation field to improve information quality. It is based on extensive research and experience, including Robert Horn's Information Mapping and Hughes Aircraft's STOP (Sequential Thematic Organization of Proposals) technique. Note that many DITA topic types are not necessarily closely connected with traditional Information Mapping.

Information typing is a practice designed to keep documentation focused and modular, thus making it clearer to readers, easier to search and navigate, and more suitable for reuse. Classifying information by type helps authors perform the following tasks:

- Develop new information more consistently
- Ensure that the correct structure is used for closely related kinds of information (retrieval-oriented structures like tables for reference information and simple sequences of steps for task information)
- Avoid mixing content types, thereby losing reader focus
- Separate supporting concept and reference information from tasks, so that users can read the supporting information if needed and ignore if it is not needed
- Eliminate unimportant or redundant detail
- Identify common and reusable subject matter

DITA currently defines a small set of well-established information types that reflects common practices in certain business domains, for example, technical communication and instruction and assessment. However, the set of possible information types is unbounded. Through the mechanism of specialization, new information types can be defined as specializations of the base topic type (`<topic>`) or as refinements of existing topics types, for example, `<concept>`, `<task>`, `<reference>`, or `<learningContent>`.

You need not use any of the currently-defined information types. However, where a currently-defined information type matches the information type of your content, use the currently-defined information type, either directly, or as a base for specialization. For example, for information that is procedural in nature, use the task information type or a specialization of task. Consistent use of established information types helps ensure smooth interchange and interoperability of DITA content.

1.3.5 Generic topics

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

For authors, typed content is preferred to support consistency in writing and presentation to readers. The generic topic type is best used only if authors are not trained in information typing or when a specialized topic type is inappropriate. The OASIS DITA standard provides several specialized topic types, including concept, task, and reference that are critical for technical content development.

For those pursuing specialization, specialize new topic types from appropriate ancestors to meet authoring and output requirements.

Comment by Kristen James Eberlein on 08 august 2023
This topic needs to go away. In general, I think many of these topics about topics are enforcing a techcomm perspective on DITA best practices, and I don't really think we want to perpetuate that here in the spec.
Disposition: Unassigned

Comment by Nancy Harrison
I agree; I think my suggestion to the 'information typing' section in 'Basic concepts' obviates any necessity for this topic
Disposition: Unassigned

1.3.6 Topic structure

All topics have the same basic structure, regardless of topic type: title, description or abstract, prolog, body, related links, and nested topics.

All DITA topics must have an XML identifier (the `@id` attribute) and a title. The basic topic structure consists of the following parts, some of which are optional:

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

Title
The title contains the subject of the topic.

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

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 are not required, they can make a dramatic difference to the usability of an information set and should generally be provided for all topics.

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

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

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

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 shells. 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 re-usable 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 re-usable 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.

1.3.7 Topic content

The content of all topics, regardless of topic type, is built on the same common structures.

Topic body

The topic body contains all content except for that contained in the title or the short description/ 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.

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 recommend 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.

<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. For content that requires a title, use `<section>` or `<example>`.

<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>`.

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.

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.

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.

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.

1.4 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.

1.4.1 Definition of 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.

Maps draw on a rich set of existing best practices and standards for defining information models, such as hierarchical task analysis. They also support the definition of non-hierarchical relationships, such as matrices and groups, which provide a set of capabilities that has similarities to Resource Description Framework (RDF) and ISO topic maps.

DITA maps use `<topicref>` elements to reference DITA topics, DITA maps, and non-DITA resources, for example, HTML and TXT files. The `<topicref>` elements can be nested or grouped to create relationships among the referenced topics, maps, and non-DITA files; the `<topicref>` elements can be organized into hierarchies in order to represent a specific order of navigation or presentation.

DITA maps impose an architecture on a set of topics. Information architects can use DITA maps to specify what DITA topics are needed to support a given set of user goals and requirements; the sequential order of the topics; and the relationships that exist among those topics. Because DITA maps provide this context for topics, the topics themselves can be relatively context-free; they can be used and reused in multiple different contexts.

DITA maps often represent a single deliverable, for example, a specific Web site, a printed publication, or the online help for a product. DITA maps also can be subcomponents for a single deliverable, for example, a DITA map might contain the content for a chapter in a printed publication or the troubleshooting information for an online help system. The DITA specification provides specialized map types; book maps represent printed publications, and subject scheme maps represent taxonomic or ontological classifications. However, these map types are only a starter set of map types reflecting well-defined requirements.

Comment by robander on 7 April 2023
Updated to remove "learning" as a type of map the spec provides.

With tech comm becoming a separate spec, is it really correct to say "The DITA specification provides...book maps"? We could say "The DITA specifications provide" (plural)? That seems simpler for an overview topic than trying to explain "This package has one specialization and another package [out later] has book maps"

Disposition: Unassigned

DITA maps establish relationships through the nesting of `<topicref>` elements and the application of the `@collection-type` attribute. Relationship tables also can be used to associate topics with each other based on membership in the same row; for example, task topics can be associated with supporting concept and reference topics by placing each group in cells of the same row. During processing, these relationships can be rendered in different ways, although they typically result in lists of "Related topics" or "For more information" links. Like many aspects of DITA, the details about how such linking relationships are presented is determined by the DITA processor.

DITA maps also define keys and organize the contexts (*key scopes*) in which key references are resolved.

1.4.2 Purpose of DITA maps

DITA maps enable the scalable reuse of content across multiple contexts. They can be used by information architects, writers, and publishers to plan, develop, and deliver content.

DITA maps support the following uses:

Defining an information architecture
Maps can be used to define the topics that are required for a particular audience, even before the topics themselves exist. DITA maps can aggregate multiple topics for a single deliverable.

Defining what topics to build for a particular output
Maps reference topics that are included in output processing. Information architects, authors, and publishers can use maps to specify a set of topics that are processed at the same time, instead of processing each topic individually. In this way, a DITA map can serve as a manifest or bill of materials.

Defining navigation
Maps can define the online navigation or table of contents for a deliverable.

Defining related links
Maps define relationships among the topics they reference. These relationships are defined by the nesting of elements in the DITA map, relationship tables, and the use of elements on which the `@collection-type` attribute is set. On output, these relationships might be expressed as related links or the hierarchy of a table of contents (TOC).

Defining an authoring context
The DITA map can define the authoring framework, providing a starting point for authoring new topics and integrating existing ones.

Defining keys and key scopes
Maps can define keys, which provide an indirect addressing mechanism that enhances portability of content. The keys are defined by `<topicref>` elements or specializations of `<topicref>` elements, such as `<keydef>`. The `<keydef>` element is a convenience element; it is a specialized type of a `<topicref>` element with the following attributes:

- A required `@keys` attribute
- A `@processing-role` attribute with a default value of "resource-only".

Maps also define the context or contexts for resolving key-based references, such as elements that specify the `@keyref` or `@conkeyref` attribute. Elements within a map structure that specify a `@keyscope` attribute create a new context for key reference resolution. Key references within such elements are resolved against the set of effective key definitions for that scope.

Specialized maps can provide additional semantics beyond those of organization, linking, and indirection. For example, the subjectScheme map specialization adds the semantics of taxonomy and ontology definition.

1.4.3 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.

Comment by rodaande
We currently redefine a lot of attributes in this topic that are more comprehensively defined in the element reference; we need to reconcile those that are defined differently and ideally reuse definitions.

Kris Eberlein, 28 September 2022

I alphabeticized the attributes in this topic. I also added them to draft comments in the definitions in the "Attributes" topics, so that we could consider them side-by-side.

Disposition: Unassigned

DITA maps often encode structures that are specific to a particular medium or output, for example, Web pages or a PDF document. Attributes, such as `@deliveryTarget` and `@toc`, are designed to help processors interpret the DITA map for each kind of output.

Comment by Kristen J Eberlein on 04 July 2019

The following paragraph seems off ...

Disposition: Unassigned

Many of the following attributes are not available in DITA topics; individual topics, once separated from the high-level structures and dependencies associated with a particular kind of output, should be entirely reusable regardless of the intended output format.

@cascade

Specifies whether the default rules for the cascading of metadata attributes in a DITA map apply. The following values are specified:

merge
Indicates that the metadata attributes cascade, and that the values of the metadata attributes are additive. This is the processing default for the `@cascade` attribute.

nomerge
Indicates that the metadata attributes cascade, but that 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 ancestor elements.

For more information, see 1.4.4.4 Example: How the cascade attribute functions (20).

@chunk
Specifies that the processor generates an interim set of DITA topics that are used as the input for the final processing. This 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.

@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/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, the behavior of the attribute is undefined.

@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.

@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 participates 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.

@processing-role

Specifies whether the topic or map referenced is processed normally or treated as a resource that is only included in order 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 is 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.

@search

Specifies whether the topic is included in search indexes.

@toc

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

Attributes in the list above are used exclusively or primarily in maps, but many important map attributes are shared with elements in topics. DITA maps also use many of the following attributes that are used with linking elements in DITA topics, such as `<link>` and `<xref>`:

- `@format`
- `@href`
- `@keyref`
- `@scope`
- `@type`

The following metadata and reuse attributes are used by both DITA maps and DITA topics:

- `@rev`, `@status`, `@importance`
- `@dir`, `@xml:lang`, `@translate`
- `@id`, `@conref`, `@conrefend`, `@conkeyref`, `@conaction`
- `@props` and any attribute specialized from `@props`, including those integrated by default in the OASIS-provided document-type shells: `@audience`, `@deliveryTarget`, `@platform`, `@product`, `@otherprops`
- `@search`

When new attributes are specialized from `@props` or `@base` as a domain, they can be incorporated into both map and topic structural types.

1.4.4 Examples of DITA maps

This section of the specification contains simple examples of DITA maps. The examples illustrate a few of the ways that DITA maps are used.

1.4.4.1 Example: DITA map that references a subordinate map

This example illustrates how one map can reference a subordinate map using either `<mapref>` or the basic `<topicref>` element.

The following code sample illustrates how a DITA map can use the specialized `<mapref>` element to reference another DITA map:

```
<map>
  <title>DITA work at OASIS</title>
  <topicref href="oasis-dita-technical-committees.dita">
    <topicref href="dita_technical_committee.dita"/>
    <topicref href="dita_adoption_technical_committee.dita"/>
  </topicref>
  <mapref href="oasis-processes.ditamap"/>
  <!-- ... -->
</map>
```

The `<mapref>` element is a specialized `<topicref>` intended to make it easier to reference another map; use of `<mapref>` is not required for this task. This map also could be tagged in the following way:

```
<map>
  <title>DITA work at OASIS</title>
  <topicref href="oasis-dita-technical-committees.dita">
    <topicref href="dita_technical_committee.dita"/>
    <topicref href="dita_adoption_technical_committee.dita"/>
  </topicref>
<topicref href="oasis-processes.ditamap" format="ditamap"/>
<!-- ... -->
</map>
```

With either of the above examples, during processing, the map is resolved in the following way:

```
<map>
  <title>DITA work at OASIS</title>
  <topicref href="oasis-dita-technical-committees.dita">
    <topicref href="dita_technical_committee.dita"/>
    <topicref href="dita_adoption_technical_committee.dita"/>
  </topicref>
  <!-- Contents of the oasis-processes.ditamap file -->
  <topicref href="oasis-processes.dita">
    <!-- ... -->
  </topicref>
  <!-- ... -->
</map>
```

1.4.4.2 Example: DITA map with a simple relationship table

This example illustrates how to interpret a basic three-column relationship table used to maintain links between concept, task, and reference material.

The following example contains the markup for a simple relationship table:

```
<map>
<!-- ... -->
<reltable>
  <relheader>
    <relcolspec type="concept"/>
    <relcolspec type="task"/>
    <relcolspec type="reference"/>
  </relheader>
  <relrow>
    <relcell>
      <topicref href="A.dita"/>
    </relcell>
    <relcell>
      <topicref href="B.dita"/>
    </relcell>
    <relcell>
      <topicref href="C1.dita"/>
      <topicref href="C2.dita"/>
    </relcell>
  </relrow>
</reltable>
</map>
```

A DITA-aware tool might represent the relationship table graphically:

type="concept"	type="task"	type="reference"
A	B	C1 C2

When the output is generated, the topics contain the following linkage:

A

Links to B, C1, and C2

B

Links to A, C1, and C2

C1, C2

Links to A and B

1.4.4.3 Example: How the @collection-type and @linking attributes determine links

In this scenario, a simple map establishes basic hierarchical and relationship table links. The `@collection-type` and `@linking` attributes are then added to modify how links are generated.

The following example illustrates how linkage is defined in a DITA map:

Figure 1: Simple linking example

```
<topicref href="A.dita" collection-type="sequence">
  <topicref href="A1.dita"/>
  <topicref href="A2.dita"/>
</topicref>
<reltable>
  <relrow>
    <relcell><topicref href="A.dita"/></relcell>
    <relcell><topicref href="B.dita"/></relcell>
  </relrow>
</reltable>
```

When the output is generated, the topics contain the following linkage. Sequential (next/previous) links between A1 and A2 are present because of the `@collection-type` attribute on the parent:

A

Links to A1, A2 as children
Links to B as related

A1

Links to A as a parent
Links to A2 as next in the sequence

A2

Links to A as a parent
Links to A1 as previous in the sequence

B

Links to A as related

The following example illustrates how setting the `@linking` attribute can change the default behavior:

Figure 2: Linking example with the @linking attribute

```
<topicref href="A.dita" collection-type="sequence">
  <topicref href="B.dita" linking="none"/>
  <topicref href="A1.dita"/>
  <topicref href="A2.dita"/>
</topicref>
<reltable>
  <relrow>
    <relcell><topicref href="A.dita"/></relcell>
    <relcell linking="sourceonly"><topicref href="B.dita"/></relcell>
  </relrow>
</reltable>
```

When the output is generated, the topics contain the following linkage:

A

Links to A1, A2 as children
Does not link to B as a child or related topic

A1

Links to A as a parent
Links to A2 as next in the sequence
Does not link to B as previous in the sequence

A2

Links to A as a parent
Links to A1 as previous in the sequence

B

Links to A as a related topic

1.4.4.4 Example: How the @cascade attribute functions

The following example illustrates how the `@cascade` attribute can be used to fine tune how the values for the `@platform` attribute apply to topics referenced in a DITA map.

Here a DITA map contains a collection of topics that apply to Windows, Linux, and Macintosh OS; it also contains a topic that is only applicable to users running the application on Linux.

```
<map product="PuffinTracker" platform="win linux mac" cascade="nomerge">
  <title>Puffin Tracking Software</title>
  <topicref href="introduction.dita"/>
  <topicref href="setting-up-the-product.dita"/>
  <topicref href="linux-instructions.dita" platform="linux"/>
</map>
```

The values of the `@platform` attribute set at the map level cascade throughout the map and apply to the `introduction.dita` and `setting-up-the-product.dita` topics. However, since the value of the `@cascade` attribute is set to "nomerge", the value of the `@platform` attribute for the `linux-instructions.dita` topic does not merge with the values that cascade from above in the DITA map. The effective value of the `@platform` attribute for `linux-instructions.dita` is "linux".

The same results are produced by the following mark-up:

```
<map product="PuffinTracker" platform="win linux mac">
  <title>Puffin Tracking Software</title>
  <topicref href="introduction.dita"/>
  <topicref href="setting-up-the-product.dita"/>
  <topicref href="linux-instructions.dita" platform="linux" cascade="nomerge"/>
</map>
```

Index