Frequently Asked Questions about the Darwin Information Typing Architecture

Answers about the XML-based Darwin Information Typing Architecture (DITA) for documentation

Don Day (dond@us.ibm.com)
IBM Corporation

Michael Priestley
IBM Corporation

Gretchen Hargis
IBM Corporation

DITA experts Don Day, Michael Priestley, and Gretchen Hargis address the topic architecture of DITA, tips and techniques, and general DITA questions.

General DITA questions

- Why is "Darwin" in the name of this architecture?
- Where can I learn more about topic-oriented writing and user assistance?
- How does DITA differ from DocBook?
- How will changes to the DTD be made and controlled?
- May I use this DTD in my own company?
- Is DITA integrated into any IBM products?
- Is there an XML schema for the DITA DTDs?

The topic architecture of DITA

- What is a topic?
- Why topics?
- What is the topic structure in the architecture?
- What is progressive disclosure in a topic?
- Can topics be nested?
- What is an information type?
- Why information types?
• What is specialization?

Tips and techniques
• How can I combine several topics into a single document?
• What if my information doesn't break down into topics?
• When should I specialize?
• How do I specialize?
• How do I extend specialization-aware transforms?
• May I use HTML in this DTD?
• Where can I see the DITA DTD in use?

General DITA questions
Q: Why is "Darwin" in the name of this architecture?
A: The entire name of the architecture has this combined explanation:

• Darwin: it uses the principles of specialization and inheritance
• Information Typing: it capitalizes on the semantics of topics (concept, task, reference) and of
  content (messages, typed phrases, semantic tables)
• Architecture: it provides vertical headroom (new applications) and edgewise extension
  (specialization into new types) for information

This architecture supports the proper construction of specialized DTDs from any higher-level
DTD or schema. The base DTD is database DTD, which contains an archetype topic structure
and three additional peer topics that are typed specializations from the basic topic: concept, task,
and reftopic. The principles of specialization and inheritance resemble the principle of variation in
species proposed by Charles Darwin. So the name reminds us of the key extensibility mechanism
inherent in the architecture.

Return to Top

Q: Where can I learn more about topic-oriented writing and user assistance?
A: Look over the topic architecture FAQs below, and then try the following two introductory sites on
information architectures:

• Argus Center for Information Architecture
• 10 Questions about Information Architecture

Return to Top

Q: How does DITA differ from DocBook?
A: It's important to recognize that DocBook and DITA take fundamentally different approaches.
DocBook was originally designed for a single, continuous technical narrative (where the narrative might be of article, book, or multi-volume length). Through transforms, DocBook can chunk this technical narrative into topics to provide support for Web sites and other information sets. Because the goal of the DocBook DTD is to handle all standard requirements for technical documentation, the usage model encourages customization to exclude elements that aren’t local requirements. The usage model supports but discourages local extensions because of the potential for unknown new elements to break tool support and interoperability.

By contrast, DITA was designed for discrete technical topics. DITA collects topics into information sets, potentially using filtering criteria. The core DITA information types are not intended to cover all requirements but, instead, provide a base for meeting new requirements through extension. Extension is encouraged, but new elements must be recognizable as specializations of existing elements. Through generalization, DITA provides for tool reuse and interoperability.

Each approach has its strengths. DocBook would be the likely choice for a technical narrative. DITA would be the likely choice for large, complex collections of topics or for applications that require both extensibility and interoperability. Technical communications groups might want to experiment with both packages to determine which approach is better suited for their processes and outputs.

Q: How will changes to the DTD be made and controlled?

A: The Darwin Information Typing Architecture was first introduced in April 2001. Since then users have discussed issues about the DITA within IBM and on the DITA forum, and various changes have evolved, leading to a major refresh a year later. The design will slow down so that the interested user community -- you! -- can focus on learning about and using DITA.

Use the DITA forum to discuss the use of the DITA DTDs and style sheets. The read-me document lists several known limitations, but doubtless others await discovery as you use the DTDs. Through discussion in the DITA forum, the significant ideas will be identified and applied to subsequent refreshes of the package. The forum will be actively monitored by the DITA project’s architects, Don Day and Michael Priestley, among others.

Q: May I use this DTD in my own company?

A: Yes, we encourage you to use it.
Q: Is DITA integrated into any IBM products?
A: Yes. We have several projects underway that are using DITA. The purpose of those projects is to continue to validate the DITA architecture and use the DTD in a product development environment.

Return to Top

Q: Is there an XML schema for the DITA DTDs?
A: Yes, the DITA toolkit provides both DTD and XML Schema representations of the architecture. The basic concepts of DITA are not tied to implementation. Both schemas and DTDs can be used to define specializable DITA elements.

Return to Top

The topic architecture of DITA

Q: What is a topic?
A: A topic is a chunk of information organized around a single subject. Structurally, it is a title followed by text and images, optionally organized into sections. Topics can be of many different types, the most common being concepts, tasks, and reference.

Return to Top

Q: Why topics?
A: DITA is based on topics because they are the optimal size to allow reuse in different delivery contexts without affecting a writer’s efficiency. If we choose a smaller unit, the writer needs to check the unit in all its contexts to make sure that information flows correctly. If we choose a larger unit, the information cannot be easily reassembled into structures that different delivery contexts (such as a Web site or a book) require. A topic is large enough to be self contained from a writer’s point of view but small enough to reuse effectively in whatever higher-level structure a particular delivery context requires.

Return to Top

Q: What is the topic structure in the architecture?
A: The topic structure is the result of some conditions that we established for the document architecture:

• `<topic>` is the container for a single non-nesting body and any number of nesting topics.
• `<title>` provides self-description, consistent with guidelines for authoring.
• `<body>` is the container for paragraph-level content and any number of non-nesting sections.
• A topic can be augmented by a prolog, a short description, and other optional metadata.

These conditions lead to the following structure:

```
<!ELEMENT topic (title, titlealts?, shortdesc?, prolog?, body,
                 related-links?, (%info-types;)* )>
```

See the Sample topic, and its explanation, The structure of a DITA topic.

Return to Top

**Q: What is "progressive disclosure" in a topic?**

**A:** Because each topic has a title and short description in addition to its full content, applications can provide progressive disclosure. For example, a user can hover over a link to see its short description and then decide whether to follow the link for the rest of the topic. Progressive disclosure also allows topics to be meaningfully browsed in a variety of viewing contexts, whether full-screen browsers, integrated help panes, infopops, or PDA screens. The application can disclose as much information as the context supports, letting the user decide where and how to drill down to more content.

Return to Top

**Q: Can topics be nested?**

**A:** Topics can be nested to create larger document structures. However, the nesting always occurs outside the content boundary, so that child and parent topics can be easily separated and reused in different contexts (see The structure of a DITA topic). Here is a sample nesting structure:

```
<topic>
<title>A general topic</title>
  <shortdesc>This general topic is pretty general.</shortdesc>
  <body><p>General topics are not very specific. They are useful for the big picture, but they don't get into details in the same way as more specific topics.</p></body>
</topic>

<title>A specific topic</title>
  <shortdesc>This is a more specific topic.</shortdesc>
  <body><p>Specifically, this is more specific.</p></body>
</topic>
```

You can author topics either as nested structures or as individual stand-alone documents. In the latter case, you assemble the documents into nested structures as required, such as when delivering printed or printable information that has a part and chapter hierarchy.
The nested structure gives a sequence and hierarchy of topics within a topic collection. In a Web environment you could disassemble this structure into individual topics and preserve the hierarchy in a generated navigation map or table of contents. However, if the Web is the main delivery vehicle, you might want to author the topics as separate documents and then apply several tables of contents to the same collection of topics.

Q: What is an information type?

A: An information type describes a category of topics, such as concepts, tasks, or reference. Typically, different information types support different kinds of content. For example, a task typically has a set of steps, whereas a reference topic has a set of customary sections, such as syntax, properties, and usage.

Q: Why information types?

A: With information types, you can divide topics into categories that you can manage and keep consistent more easily than without information types. Information types also make it easier for users to find the information that they are looking for: how-to information in a task versus background information in a concept versus detailed specifications in a reference topic.

Q: What is specialization?

A: Specialization is the process of creating new categories of topics, or information types, as well as new categories of elements, or domain types. You can define these new types using the existing ones as a base. For example, a product group might identify three main types of reference topic -- messages, utilities, and APIs -- and define three domains -- networking, programming, and databases. By creating a specialized topic type for each kind of reference information, and creating a domain type for each kind of subject, the product architect can ensure that each type of topic has the appropriate structures and content. In addition, the specialized topics make XML-aware search more useful, because users can make fine-grained distinctions. For example, a user could search for $xyz$ only in messages or only in APIs, as well as searching for $xyz$ across reference topics in general.

Rules govern how to specialize safely: Each new information type must map to an existing one, and new information types must be more restrictive than the existing one in the content that they allow. With such specialization, new information types can use generic processing streams for translation, print, and Web publishing. Although a product group can override or extend these
processes, they get the full range of existing processes by default, without any extra work or maintenance. The DITA specialization articles outline the rules for each kind of specialization (topic type and domain type).

**Tips and techniques**

**Q: How can I combine several topics into a single document?**

**A:** The DITA design has a unified content reuse mechanism by which an element can replace itself with the content of a like element elsewhere, either in the current topic or in a separate topic that shares the same content models. The distinction between reusable content and reusing content, which is enshrined in the file entity scheme, disappears: Any element with an ID, in any DITA topic, is reusable by conref.

DITA’s conref "transclusion" mechanism is similar to the SGML conref mechanism, which uses an empty element as a reference to a complete element elsewhere. However, DITA requires that at least a minimal content model for the referencing element be present, and performs checks during processing to ensure that the replacement element is valid in its new context. This mechanism goes beyond standard XInclude, in that content can be incorporated only when it is equivalent: If there is a mismatch between the reusing and reused element types, the conref is not resolved. It also goes beyond standard entity reuse, in that it allows the reused content to be in a valid XML file with a DTD. The net result is that reused content gets validated at authoring time, rather than at reuse time, catching problems at their source.

Content referencing can be used at any scope of elements in a DITA document, from a keyword phrase that contains only PCDATA to a whole topic with other nested topics. Conref can cross file boundaries, using the same syntax as that of the href attribute on the xref element. If your authoring DTD allows topic nesting, you can create a set of minimal child topics and then use their conref attributes to pull in content from fully populated topics in other files.

**Q: What if my information doesn't break down into topics?**

**A:** Most information can be broken down into topics (headings and content). However, if your information requires a more seamless flow of information across topic boundaries, don't use this architecture.
**A:** Create specialized topics when you have a restrictive category of topics that you want to keep consistent and that your users want to distinguish from other categories. Create specialized domains when you have a set of elements that you want available across several of your topic types. Be sure to specialize from the correct base: For example, categories of reference topics should specialize `<reference>`, categories of tasks should specialize `<task>`, and domain types should always specialize either `<topic>` or another domain type. If you need to allow more content structures than the base types allow, you can specialize directly from topic, or form your own base type. However, the lower down in the hierarchy that you can specialize, the better; you can then take advantage of any transforms or processes that have been developed for the more general categories that you specialize from.

**Return to Top**

**Q: How do I specialize?**

**A:** You need to identify the differences between your new type of information and the more general type that you are specializing from. After you have identified the differences, you create a DTD file to declare the new elements that you require. Create another module to declare a set of mapping attributes for the new elements that point to the generic element types that they specialize. Then add import statements in the DTD file to bring in the mapping module and any ancestor modules. Finally, add a line that redefines the information types entity to include your new type. You now have a customized DTD.

When you specialize a domain, you need to first determine what elements must be specialized for the domain. Then you write an entity declaration file to list the specialized elements, along with their topic types and domain types. Next, you create a file where you define both the elements that are introduced for the domain and the specialization hierarchy. Finally, you write the shell DTD to combine the domain with topics and other domains.

These processes are described in more detail in the documents [Specializing topics in DITA](#) and [Specializing domains in DITA](#).

**Return to Top**

**Q: How do I extend specialization-aware transforms?**

**A:** See the article on [specializing topic types](#).

**Return to Top**

**Q: Can I use HTML in this DTD?**
A: Yes. Many writers have had at least some experience with HTML as a markup language. Therefore the base DITA DTD incorporates as many HTML elements as are useful for the type of technical information for which topics might be used. In addition, we have defined a subset of XHTML for which there is a very simple transformation into the DITA format -- often with no change to many content elements! In fact, if you can load an XHTML document into the same editor as an XML DITA document, you can probably copy and paste long stretches of the XHTML content directly into the topic. Regardless, to gain the real advantage of XML, you should use the semantics of the DTD.

Q: Where can I see the DITA DTD in use?

A: Right here! The original documents that accompany this proposal were authored in XML using the ditabase DTD.
Resources

- For latest information about the ongoing status of the DITA standard and its community of users, see the DITA Focus Area at dita.xml.org.
- IBM donated DITA to the OASIS standards organization in March of 2004, where it is now managed by the OASIS DITA Technical Committee (http://www.oasis-open.org/committees/dita/). In April of 2005, OASIS approved Version 1.0 of the DITA specification, which consists of the following documents:
  - A consolidated .zip file with all specifications, DTDs, and Schemas is publicly available in the documents section of the OASIS DITA Technical Committee site: http://www.oasis-open.org/committees/download.php/12091/cd2.zip

A reference implementation toolkit for both the developerWorks and OASIS 1.0 versions of the DITA DTDs/Schemas is available at the DITA Open Toolkit project site on SourceForge: http://dita-ot.sourceforge.net. The DITA Open Toolkit supersedes all previous versions published on developerWorks, the last version of which was commonly called "dita132".

- Read up on DITA in the Introduction to the Darwin Information Typing Architecture (developerWorks, updated May 2002) and in the Specializing topic types in DITA article (developerWorks, updated September 2005).
- Read Erik Hennum’s article Specializing domains in DITA, which shows you how to leverage the extensible DITA DTD to describe new domains of information
- Find out how to join the discussion in the DITA forum, moderated by Don Day and Michael Priestley.
- Go directly to the DITA forum.
- Download the latest DITA DTDs, stylesheets, and sample documents.
About the authors

Don Day

Besides his main work as husband, father, and cat lover, Don designs and supports publishing tools for IBM's Information Development community and has represented IBM on the W3C XSL and CSS Working Groups. He has B.A.s in English and Journalism and an M.A. in Technical and Professional Communication from New Mexico State University. You can reach Don at dont@us.ibm.com.

Michael Priestley

Michael Priestley is an information developer for the IBM Toronto Software Development Laboratory. He has written numerous papers on subjects such as hypertext navigation, singlesourcing, and interfaces to dynamic documents. He is currently working on XML and XSL for help and documentation management. You can reach Michael at mprietl@ca.ibm.com.

Gretchen Hargis

Gretchen is a senior software engineer working as a technical editor on application development tools. She is an author of Developing Quality Technical Information (Prentice Hall) and has a B.A. from McGill University in Russian. You can reach Gretchen at ghargis@us.ibm.com.

© Copyright IBM Corporation 2001, 2005
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)