



# Revisiting XML Technical Vocabularies: DITA, TEI and Docbook

The earliest work I did online was with Docbook and using it to create web versions of articles and essays; it later turned to ebook creation. In 2015 I created a custom XML vocabulary and conversions to HTML and PDF, documenting the process in [a series of blog posts](#)

But when these things appear in job requirements and descriptions it's not enough to say you're familiar with them, you have to demonstrate mastery of the specific technologies. So I'm dusting the cobwebs and revisiting DITA, TEI and Docbook to explore uses and capabilities.

## DITA

DITA (Darwin Information Type Architecture) was first introduced by IBM in 2000 as an end-to-end, single source content system.

DITA defines an XML architecture for designing, writing, managing, and publishing many kinds of information in print and on the Web.

[OASIS, DITA XML Cover Pages](#)

The idea is that you author the content once and can reuse it in different publications for both print and the web.

DITA is based on the concept of modules. As of version 1.3 (the latest OASIS standard version), DITA include a generic topic type and five specializations of the generic topic:

- A **Task** topic is intended for a procedure that describes how to accomplish a task
  - It lists a series of steps that users follow to produce an intended outcome
- **Concept** information is more objective, containing definitions, rules, and

guidelines.

- A **Reference** topic is for topics that describe command syntax, programming instructions, and other reference material
- A **Glossary Entry** topic is used for defining a single definition for a given term
- The **Troubleshooting topic** describes troubleshooting scenario, listing both a problem followed by one or more cause/solution(s) sets.

In addition to the generic topics DITA allows for the creation of even more specialized content types via **specializations**, subtypes of the defaults that serve specific purposes.

The following subcommittees are officially part of the OASIS DITA Technical Committee.

- [DITA Help Subcommittee](#)
- [DITA Learning and Training Content Specialization Subcommittee](#)
- [DITA Technical Communication Subcommittee](#)
- [Lightweight DITA Subcommittee](#)

These are important because each subcommittee may have it's own DITA based vocabulary to match their needs.

One additional content that is important is the map. In the context of DITA, maps are documents that organize topics and associated resources into structured collections of information. DITA maps specify hierarchy and the relationships among the topics; they also provide the context in which keys are defined and resolved. DITA maps should have `.ditamap` file extensions.

DITA maps use `<topicref>` elements (or specializations of `<topicref>`) 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 between 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 the topics needed to support a set of user requirements, the sequence of the topics; and the relationships between the topics.

Because DITA maps provide a context for topics, the topics themselves can be

relatively context-free and can be used and reused in multiple different contexts.

DITA maps often represent a single deliverable, like a Web site, a printed publication, or the online help for a product. They also can be subcomponents for a deliverable, for example, a chapter in a printed publication or a section of a website.

The DITA specification provides specialized map types; book maps represent printed publications, subject scheme maps represent taxonomic or ontological classifications, and learning maps represent formal units of instruction and assessment.

# Docbook

[DocBook](#) is a semantic markup language for technical documentation. It was originally intended for writing technical documents related to computer hardware and software, it has evolved into a general purpose authoring system.

Docbook's vocabulary is presentation-agnostic. We can use the same document to create HTML pages, PDF documents and Epub ebooks without requiring any changes to the source.

Since version 5.1, DocBook supports topic based authoring using [Assemblies](#). I've covered DocBook Assemblies when discussing

# TEI

## Structured Content Tools

### Authoring

OxygenXML

DITA Open Toolkit

Frame Maker & Madcap Flare

### Bibliography

- [Darwin Information Typing Architecture](#) — Wikipedia
- [What is DITA](#) — Techwhirl
- [Introduction to the Darwin Information Typing Architecture](#)
- [DITA 1.3: Why Three Editions? Version 1.0](#) — Oasis DITA TC
- [https://www.oxygenxml.com/dita/styleguide/webhelp-feedback/Artefact/Authoring\\_Concepts/c\\_Introduction\\_to\\_DITA.html](https://www.oxygenxml.com/dita/styleguide/webhelp-feedback/Artefact/Authoring_Concepts/c_Introduction_to_DITA.html)
- <https://www.xmlmind.com/tutorials/DITA/index.html>
- <https://learningsolutionsmag.com/articles/524/what-is-dita-and-why-should-you-care>
- <https://medium.com/technical-writing-is-easy/technical-writing-and-dita-9412deb9643>
- <https://straygoat.co.uk/topic-based-documentation/>
- <https://www.dita-ot.org/>
- <http://dita.xml.org/>