Writing topic-based content

Topic-based documentation is content broken down into topics; small reusable chunks of content that can be reused in multiple documents.

I try my best to write as if the document was topic-based. Some of the advantages I see:

- Topic-based authoring makes you think about the content you write. The content is usually more focused
- Users can go straight to the section with the information they need, and it should make sense on its own with little or no outside references
- It allows you to tailor material for different audiences
- Most topic-based authoring tools let you use one source for multiple content outputs
- It encourages teams to work together. Since we are working with chunks of a document we can assign them to different writers and editors

So what might it look like?

Let's take a technical example: We're writing docs for 4 different applications, they all share common functionality and have their own specific variances from the norm.

SO what do we do?

We write all the common topics first. These will be used in all four appliance documentation models since they describe common processes and we don't want to rewrite it for every model.

- The introduction is identical for all products, describing the product line and what it's intended to do as a family.
- The basic process for configuring an appliance is a topic common to all appliances. They all need the same network configuration and basic setup.
- We write a basic technical troubleshooting topic for the aspects that are common to all appliances.
- We write individual topics for elements that specific to each appliance. We

could break down each topic into subtopics to break down information

Once we have all the topics written we can merge them into larger documents by copying them into the large document file or by including them via links or XML tools like xinclude or similar transclusions standards and tools.

Using Docbook 5.1 as an example, we can create topic-based documentation using Assemblies composed of the following elements:

- assembly is the root element for Docbook Assembly documents
- resources acts as a container for resources managed and used in this assembly
- resource represents individual resources indicated by URLs
- structure pulls content from one or more topics
- module represents one component of a structure
- topic is a semantically neutral unit of content. In the context of assemblies, it is used in addition to other units of content available in Docbook

The `assembly` document looks like this:

```
<assembly
  version="5.1"
  xmlns="http://docbook.org/ns/docbook">
  <resources>
    <resource
      href="topic1.xml"
      xml:id="topic1"/>
    <resource
      href="topic2.xml"
      xml:id="topic2"/>
    <resource
      href="topic3.xml"
      xml:id="topic3"/>
  </resources>

  <structure>
    <module resourceref="topic1"/>
    <module resourceref="topic2"/>
    <module resourceref="topic3"/>
```

```
  </structure>
</assembly>
```

And a topic document, in this case `topic1.xml`, could look like this:

```
<topic xmlns='http://docbook.org/ns/docbook' version="5.1">
<info>
  <title>Topic 1</title>
</info>

<para>This is the content for Topic 1.</para>

</topic>
```

We could also use chapter, section, bibliography, glossary or any other element that we'd normally use as the root of a document, including those elements listed in RFE 1899655.

Links and resources

- Creating DocBook Documents In Docbook 5.1: The definitive guide. It provides an overview of how to write Docbook content
- Assemblies In Docbook 5.1: The definitive guide. Covers Docbook assemblies, how they work, and additional areas that I did not cover in the post
- XML Mind Docbook assemblies tutorial