



DITA hands-on

Carlos Evia (cevia@vt.edu)

Today's menu

- DITA topics review
- DITA maps
- Single sourcing
- Tools to author and process DITA
- Hands-on DITA work
- Conrefs
- Publication/transformation.

DITA topics review

- Concept
- Task
- Reference
 - Check your sidekick!



Concept

- Answers a “what” question
- Good for explanations/relationships/descriptions/introductions

```
<concept id="cake">  
<conbody>  
<title>This is a cake</title>  
<p>Cakes are delicious things.</p>  
</conbody>  
</concept>
```

Task

- Answers a “how to” question
- Good for instructions and procedures

```
<task id="eat">  
<taskbody>  
<title>How to eat a cake.</title>  
<steps>..... </steps>  
</taskbody>  
</task>
```

Reference

- Answers a “where” question
- Good for glossaries, archives, collections.

```
<reference id="types">  
<refbody>  
<title>Types of cake</title>  
<p>Collection of cakes.</p>  
</refbody>  
</reference>
```

DITA Maps

DITA MAP

A DITA map file is used to tie together the concepts and tasks into a single document. The file extension for a dita map is "file.ditamap" where "file" is the name you would like for the map. A DITA map needs to contain references to the locations of the topics you wish to include. The following is in the form of a guided example.

A DITA Map file starts with the file type declaration:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE map PUBLIC "-//OASIS//DTD DITA  
Map//EN" "http://docs.oasis-  
open.org/dita/v1.1/OS/dtd/map.dtd">
```

Then the map element opens and includes a title for the map:

```
<map title="Enter name the desired title of your map">
```

Topics references can be simply inline:

```
<topicref href="topics/introduction.xml" />
```

Or topic references can be nested within other topic references to create a heirarchy:

```
<topicref href="topics/overarching_topic.xml">  
  <topicref href="tasks/nested_task_name.xml">  
  </topicref>  
</topicref>
```

Then, you would want to close the map

```
</map>
```

Prologs and metadata

- No, we don't need them
- But they can help... users don't see them
- Info about author, copyright, critical dates, etc.

Audience and author

- Audience type: user, purchaser, administrator, programmer, executive, and services.
- Experience level: novice, general, and expert.
- Author: creator and contributor.

```
<prolog>
<audience type="user" experiencelevel="novice" />
<author type="creator">Eric Cartman</author>
<author type="contributor">Stan Marsh</author>
<critdates>
<created date="2010-06-12" expiry="2011-08-20" />
</critdates>
<keywords>
  <keyword>chocolate</keyword>
  <keyword>ice cream</keyword>
  <keyword>machine</keyword>
  <keyword>cleaning</keyword>
  <keyword>installing</keyword>
</keywords>
</prolog>
```

Single sourcing

- Reuse and repurpose content
- Never write a topic (or element) twice
- Example: *Topics can be assembled in different ways and produce separate manuals for new and advanced users with a few clicks.*



Map # 1

- Orange id= “cone”
- Yellow
- Red
- Orange id = “square”

Map # 2

- Yellow
- Orange id = “square”
- Blue
- Red

Tools to author DITA

- The DITA Open Toolkit (OT)
- FrameMaker (TCS)
- XMLSpy
- **oXygen**
- **XML Editor**

Hands-on DITA work

- Create a cookbook with your special theme
- You will need,
 - an introductory topic (c****t)
 - at least three recipes (t**k)
 - optional glossary of terms (r*****e)

Need help?

- Look at the monkey example
- Look at the ice cream maker example
- Use your DITA sidekick
- Ask questions

Introducing conrefs

- Content reference=Conref
- Conrefs can borrow information among topics
- From Task to Concept to Reference and any other combinations
- Big daddies of single-sourcing.

Example

- A disclaimer: Every topic should have a disclaimer saying “People are naturally dumb. Our products are not to blame if users do dumb things with them (or whatever)”
- We could
 - a) Copy and paste/rewrite disclaimer inside every topic
 - b) Use a Conref

```
<concept id="nosmart">
  <title>People no smart</title>
  <conbody>
    <p id="disclaimer">People are naturally
dumb. Our products are not to blame if
users do dumb things with them (or
whatever)</p>
  </conbody>
</concept>
```

```
<task id="smart">
  <title>How to be smart</title>
  <taskbody>
    <steps>
      <step>
        <cmd>Read (not only Harry Potter)</
cmd>
      </step>
      <step>
        <cmd>Listen to Mozart</cmd>
      </step>
    </steps>
    <postreq>
      <p conref="nosmart.xml#nosmart/
disclaimer" />
    </postreq>
```

Configuring a step

```
<step id="mildsalsa">  
  <cmd>Prepare the salsa</cmd>  
  <substeps>  
    . . . .  
  </substeps>  
</step>
```

```
<step>
    <cmd>Prepare the chips</cmd>
</step>
<step conref="salsas.xml#salsa/
mildsalsa">
    <cmd/>
</step>
```

Publication/ transformation

- DITA collections can generate deliverables as
 - PDF
 - HTML
 - WebHelp
 - ePub

Customizing outcomes

- For HTML deliverables, some good CSS will do
- For PDF deliverables, some good XSL will do (and this is a biggie).

Resources

- www.ditausers.org
- DITA users Yahoo Group
- “Practical DITA,” by J. Vazquez
- “DITA 101,” by A. Rockley et al.
- “The DITA Style Guide,” by T. Self.
- More books coming soon...