

Basic DITA & Arbortext Editor Refresher

Session 1

- DITA overview
- Basic authoring in Arbortext Editor
- Common elements and task topics
- Attributes, metadata and profiles
- Hazard statements

DITA OVERVIEW

The what and the why

- What is DITA?

- XML architecture that provides a data model for authoring, managing and publishing structured, topic-based content, with a core set of topics derived from concept, task and reference

- What are the benefits of moving to DITA & a CMS?

- Reuse & being faster
- Consistency
- Global

Key concepts

Term	Definition
Topic	<ul style="list-style-type: none">• Standalone unit of information that answers one question• Small enough to reuse across multiple contexts and output media and allow organization into a variety of content structures• Large enough to be readable, coherent and meaningful enough to stand on its own
Information type	<ul style="list-style-type: none">• The main categories for content in DITA are task, concept and reference<ul style="list-style-type: none">• Task: Answers “How do I...?”• Concept: Answers “What is this thing (feature, process, idea)?” or “Why should I...?”• Reference: Provides lookup information• You can further refine these to provide more consistency (for example, you can have a concept type of “protocol implementation”) – we haven’t done much of this yet
Map	Container for DITA topics or other maps used to organize topics into structures for management, navigation or publishing
Semantic markup	Marking up pieces of content as what they are rather than how they should look (for example, <uicontrol> rather than <bold>)

Key terms

Term	Definition	Example
Element	Unit within an XML document made up of a start tag, an end tag, the content between the tags, and any XML attributes (often used interchangeably with 'tag').	<code><section brand="Veris">section content</section></code>
Tags	The strings used to contain content and add structure the elements (often used interchangeably with 'element').	<code><section></section></code>
Attributes	<ul style="list-style-type: none">• Name/value pair that provides additional information about an element• Support content reuse, filtering and translations (among other things)	<code><image brand="Veris"></code> <code><ph translate="no"></code>
Metadata	Information about content	Author=Schneider Electric RevDate=06/2013

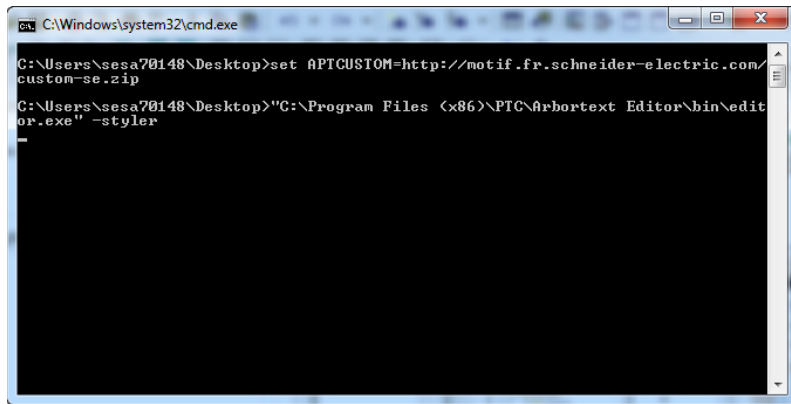
Main types of DITA elements

Type	Description	Context	Example
Block	Contain major blocks of content and provide the structure	Where you would add a hard return or use a new paragraph style in Word or FM	<section> <title> <p> (paragraph) (unordered list)
Inline	Mark up snippets or phrases of content within blocks of content	Where you would apply character formatting in Word or FM	<uicontrol> <tm> <codeph> <keyword>

BASIC AUTHORIZING IN ARBORTEXT EDITOR

Arbortext Editor interface and basic authoring

- Launching Arbortext Editor with SE configuration:
 - Always launch using the .bat file
 - Making sure the cmd file runs – you should see something like this:



```
C:\Windows\system32\cmd.exe
C:\Users\sesa70148\Desktop>set APTCUSTOM=http://motif.fr.schneider-electric.com/
custom-se.zip
C:\Users\sesa70148\Desktop>"C:\Program Files (x86)\PTC\Arbortext Editor\bin\edit
or.exe" -styler
```

- If you don't, you may need to close and try again.
- Basic AE interface overview and instructions are on Box in the Migration folder (note, these need to be updated)

COMMON ELEMENTS & TASK TOPICS

Common topic elements

Common elements across topic types

title
shortdesc
prolog
conbody, refbody, taskbody

section
p
ul, ol
dl
ph
example

fig, image
table
related-links

Common elements and their usage: [R_CommonElements.pdf](#)

Tasks and their unique elements

- “How do I do this one task (successfully)?”
 - Provides the steps and minimal related information the user requires to accomplish a single task

Elements unique to tasks

prereq (prerequisite) context steps substeps	cmd (command) stepinfo (step note) stepresult stepxmp (step example) choices / choice table	result (task result) example (task example) postreq (post-requisite)
---	---	--

Common elements and their usage: [R_TaskElements.pdf](#)

Walkthrough and practice

- Goal:
 - Create a new topic and edit it
- Exercise files:
 - Source PDF for practice exercise: [T_configuring_meter.pdf](#)
- Reference files
 - Short list of common elements across topics: [R_CommonElements.pdf](#)
 - Short list of unique task elements: [R_TaskElements.pdf](#)
- Basic AE instructions are on Box in the Migration folder (note, these need to be updated)

ATTRIBUTES, METADATA & PROFILES[†]

[†] There is a deeper dive into attributes, metadata and profiles in the next session

Attributes and Metadata

● Attributes:

- Name/value pair that provides additional information about an element
- Support content reuse and translations (among other things)
- Examples: <image **brand**="Veris">, <ph **translate**="no"> or <hazardstatement **type**="Danger">

● Metadata:

- Provide additional information about the content
- Metadata can be manage in the CMS, the map or the topic, depending on the information being added and the context in which it applies
- Examples:
 - **Author=Schneider Electric, RevDate=06/2013, indexterm=BACnet**

● Profiles:

- AE-specific method for accomplishing conditional processing (for example, generating output for alternate brands) – note that this uses a special subset of attributes
- Examples: **product=Siemens_Ind** or **audience=electrician**

Topic-level metadata

- Contained in the <prolog> element
- Most common metadata elements that you might use
 - <metadata>
 - <keywords>: contains keyword and indexterm elements
 - <audience> (though the list of audiences in the profiles may not be useful to us)
 - <prodinfo>: contains elements that provide information about what product version the topic applies to – note that this is NOT used for conditional processing
 - Resourceid: Allows for adding IDs for help file links – not in use at the moment but may be in future
- Prolog metadata that you should NOT use
 - Author: Do not use – this is covered by Windchill
 - Publisher: Do not use – this is covered at the bookmap level

Profiles

- The AE-specific mechanism for accomplishing conditional processing
 - Think variables and conditional text in FrameMaker
- Some can be applied at element or map level
 - Audience/audience
 - Brand/product
 - Publishing Target/platform
 - Access Level/otherprops
- Some should only be applied at map level (by translation vendor) except (possibly) in rare circumstances:
 - Language/xml-lang
- Some can only be applied to maps
 - Format size/format

HAZARD STATEMENTS

Hazard statement elements and metadata

- Hazard statement elements:

- hazardstatement = hazard table in FrameMaker
- typeofhazard = hazard statement (for example, Hazard of electric shock, explosion or arc flash)
- consequence = failure to follow
- howtoavoid = avoidance statement(s)

- Hazard statement attribute:

- Type: danger, warning, caution or notice

Walkthrough and practice

- Goals:

- Add metadata to a topic and set the value of an element attribute
- Add a hazard statement and set the value of its “type” attribute

- Exercise files:

- Source PDF for practice exercise: [T_configuring_meter.pdf](#)

- Reference files

- List of common attributes: [M_BasicAEInstructions.pdf](#)
- List of common topic metadata: [R_TopicMetadata.pdf](#)

- Basic AE instructions are on Box in the Migration folder (note, these need to be updated)

Post-migration samples

- Post-migration samples

- [PDF](#)

- [XML](#)

Additional information

- <http://docs.oasis-open.org/dita/v1.2/spec/DITA1.2-spec.html>
 - Full DITA 1.2 spec
- <http://dita.xml.org/>
 - Online community with information and links regarding DITA
- [Power_SchneiderElectric_InfoModelD2.pdf](#)
 - Full information model for EcoBusiness Power Monitoring, developed in consultation with ComTech Services
 - Outlines how we intend DITA elements to be used
 - NOTE: this is still in progress