Giesecke & Devrient

Docbook and DITA

Printed on 9th July 2015

1 Docbook and DITA

This plugin provides a transform from Docbook to DITA and some proof-of-concept examples of interoperability.

Important:
The transform has a number of limitations and is a proof of concept rather than a robust conversion transform.

1.1 Installing the DocBook-DITA plugin

This plugin provides a transform from Docbook to DITA and some proof-of-concept examples of interoperability.

This task has 5 steps (min. no of steps to show this can be set in task-elements.xsl).

Procedure

1) You must install the DITA Open Toolkit 1.3 or higher and run a test build to verify the installation.
2) It's also recommended that you install the DocBook tools at version 1.69.1 and the DocBook 4.4 DTDs.

   Other versions may work but haven't been tested.
3) Edit the dbdita.properties file in the plugin directory to identify the location of the DocBook tools and document types.
4) In the base directory for the DITA Open Toolkit, install the plugin.
   You execute the following command (using blackslashes instead of slashes on Windows):

   ```
   ant -f demo\dbdita\install-plugin.xml
   ```
5) Verify the plugin installed correctly with a sample build.
   You can execute the following command (again, using blackslashes instead of slashes on Windows):

   ```
   ant -f demo\dbdita\run-ant.xml docbook2DITADemo
   ```

Results

You can use the DocBook-to-DITA transform on your own DocBook content or explore the interoperability demos.

Related information

Demonstration transforms page

The examples of DocBook and DITA interoperability can be processed with demonstration transforms.

1.2 The Docbook to DITA transform

This directory provides the docbook-to-dita transform.

The Docbook-to-DITA transform uses a design pattern that might be called *a dialogue transform*. This approach makes a strict separation between the input and output logic. Generating an output element requires a conversation between the output writer and the input reader:

DITA writer The DITA writer knows how to generate DITA output. After producing an element, the writer requests the attributes and content of the element from the reader. For instance, the paragraph output rule generates a `<p>` element and then requests the id attribute and other attributes of the paragraph and then the paragraph content. Where the content is a simple sequence (for instance, as with a topic or list), the writer requests in the subelements individually.

Docbook reader The Docbook reader knows how to read Docbook input. When the DITA writer requests an attribute or content, the reader is responsible for satisfying the request from the current input context. For instance, when the DITA paragraph writer requests an id attribute, the reader might copy the id attribute from the current Docbook input element or from a nearby Docbook input element. The reader can also generate an id attribute or skip the id attribute. Similarly, when the DITA paragraph writer requests paragraph content, the reader can provide paragraph content by invoking the output rules on any of the text and subelements from the input (often but not necessarily from content of the current input element).

This directory provides the following modules:

docbook2dita.xsl This module combines the Docbook reader and DITA writer to transform Docbook input to DITA output.

dbReader.xsl This module provides the base Docbook reader. The module can be extended by other modules that override the base module to handle requests from the DITA writer in different ways.

1.2.1 DITA writer

This directory contains the DITA output writer.

The output writer provides the following rules for each output element. (In the descriptions, *moduleName* stands for a module that provides an element or attribute, *elementName* stands for an element, and *attributeName* stands for an attributeName.) Each rule consists of a mode:

***moduleName.elementName*.out** Generates the *elementName* element from the *moduleName* module. For instance, the `topic.keyword.out` rule generates a `<keyword>`

	element. The rule also fires the wrapper rules to request the attributes and content of the element.
***moduleName.elementName*.atts.in**	Provides a wrapper for requesting the attributes of the *elementName* element from *moduleName* module.
***moduleName.elementName.attributeName*.att.in**	Requests the value of *attributeName* in the context of the *elementName* element from the *moduleName* module. The default handler for this rule requests the value of *attributeName* in any context.
***attributeName*.att.in**	Requests the value of *attributeName* in any context.
***moduleName.elementName*.content.in**	Provides a wrapper for requesting the content of the *elementName* element from *moduleName* module. If the content model isn't shared, the default handler for this rule requests content from the children and text (if the content model supports mixed content) of the current input node. If the content model is shared, the default handler invokes a rule for the shared content model. For instance, elements that accept content conforming to the `words.cnt` group invoke the rule for `words.cnt.text.in`. If the content model takes a simple sequence, the default handler for this rule requests the subelements in sequence.
***moduleName.elementName*.child**	Requests the current input element as a child in the context of the *elementName* element from the *moduleName* module. The default handler for this rule requests the current input element as a child in any context.

child	Requests the current input element as a child in any context.
***moduleName.elementName*.in**	Requests the *elementName* element from the *moduleName* module. This rule gets invoked only by a content wrapper when the content model has a simple sequence.

The reader can handle a request in a specific output context or in any context as appropriate. For instance, the Docbook reader handles a context-free request for an id attribute by copying the id (if present) on the current input element to the output. In addition, however, the Docbook reader handles the request for an id attribute in a topic context by generating an id if necessary.

1.2.2 DITA writer make

This transform generates the DITA writer from the DITA schema modules

The transform first merges the DITA topic or domain type modules and then reads each type module and generates a writer transform module for the DITA topic type or domain.

Note:
This transform depends on rigorous conformance to the design pattern for DITA type modules. As a result, the transform is somewhat fragile and should be run only when modules change or new modules are added.

1.3 DITA maps and DocBook

A DITA map can assemble content or define relationships between content in processable formats.

A DITA map can nest content or define relationships between content. The content can, of course, be DITA topics to take advantage of granular design, strong typing, and extensible markup. Content can also, however, be provided in a processable format such as HTML or XML by setting the format attribute and (for more precision within the format) the type attribute.

In particular, a DITA map can compose a DocBook book or establish relationships between DocBook articles. (These alternatives were summarized at a high level in http://www.oasis-open.org/events/symposium_2006/slides/Hennum.ppt).

1.3 Book composition through a map

A DITA map can define the nesting of the divisions that make up a DocBook book.

A DITA map can be specialized to introduce elements corresponding to the DocBook logical divisions such as book, chapter, section, and so on. The specialized elements can ensure that references to divisions are nested in the proper order, for example, so that a section reference doesn't nest a chapter reference. The DITA map can then be preprocessed to generate a DocBook book and then processed with DocBook tools.

1.3 Book content by reference to a map

A DocBook book can be populated with content from to a DITA map.

This solution requires customizing DocBook by adding a single element that refers to DITA maps or topics. A DocBook book can then be preprocessed to replace the references with the result of converting the referenced DITA content to DocBook using the DITA-to-DocBook transform of the DITA Open Toolkit. This approach is described in more detail in Robert Anderson's article, "Implement a DITA publishing solution without abandoning your current publishing system investments" (see http://www-128.ibm.com/developerworks/xml/library/x-dita11/).

1.3 Article relationships through a map

A DITA map can express relationships between DocBook articles.

A DITA map can be specialized for convenience to add an element to refer to a DocBook article. The specialized map can express navigation or cross-referencing relationships between a DocBook article and other DocBook articles or DITA topics. A preprocess can then push the relationships into the DocBook articles. A light extension on the DocBook processing can then format the links during processing to XHTML.

1.4 Demonstration transforms

The examples of DocBook and DITA interoperability can be processed with demonstration transforms.

The plugin contributes the following targets to the DITA Open Toolkit:

docbook2dita	Convert DocBook divisions to DITA topics. The DITA topics can then be processed with the DITA Open Toolkit. Executed by the docbook2DITADemo target in the run-ant demo build.
docbookRelatePush	Push relationships from a DITA map into DocBook articles. The DocBook articles can then be processed with the DocBook transform tools. Executed by the docbookRelateDemo target in the run-ant demo build.
docbookMix2dita	Convert DocBook articles referenced from a DITA map to DITA. The DITA map can then be processed with the DITA Open Toolkit. Executed by the docbookMixDemo target in the run-ant demo build.
docbookCompose2docbook	Assemble a DocBook book from DocBook divisions as specified by a DITA map. The DocBook book can then be processed with the DocBook transform tools. Executed by the docbookComposeDemo target in the run-ant demo build.
docbookDitaref2docbook	Populate a DocBook book by reference to a DITA map. The DocBook book can then be processed with the DocBook transform tools.

Executed by the docbookDitarefDemo target in the run-ant demo build.

2 Installing the DocBook-DITA plugin

This plugin provides a transform from Docbook to DITA and some proof-of-concept examples of interoperability.

This task has 5 steps (min. no of steps to show this can be set in task-elements.xsl).

Procedure

1) You must install the DITA Open Toolkit 1.3 or higher and run a test build to verify the installation.
2) It's also recommended that you install the DocBook tools at version 1.69.1 and the DocBook 4.4 DTDs.

   Other versions may work but haven't been tested.
3) Edit the dbdita.properties file in the plugin directory to identify the location of the DocBook tools and document types.
4) In the base directory for the DITA Open Toolkit, install the plugin.
   You execute the following command (using blackslashes instead of slashes on Windows):

   ```
   ant -f demo\dbdita\install-plugin.xml
   ```
5) Verify the plugin installed correctly with a sample build.
   You can execute the following command (again, using blackslashes instead of slashes on Windows):

   ```
   ant -f demo\dbdita\run-ant.xml docbook2DITADemo
   ```

Results

You can use the DocBook-to-DITA transform on your own DocBook content or explore the interoperability demos.

Related information

Demonstration transforms page
The examples of DocBook and DITA interoperability can be processed with demonstration transforms.

3 Demonstration transforms

The examples of DocBook and DITA interoperability can be processed with demonstration transforms.

The plugin contributes the following targets to the DITA Open Toolkit:

docbook2dita	Convert DocBook divisions to DITA topics. The DITA topics can then be processed with the DITA Open Toolkit. Executed by the docbook2DITADemo target in the run-ant demo build.
docbookRelatePush	Push relationships from a DITA map into DocBook articles. The DocBook articles can then be processed with the DocBook transform tools. Executed by the docbookRelateDemo target in the run-ant demo build.
docbookMix2dita	Convert DocBook articles referenced from a DITA map to DITA. The DITA map can then be processed with the DITA Open Toolkit. Executed by the docbookMixDemo target in the run-ant demo build.
docbookCompose2docbook	Assemble a DocBook book from DocBook divisions as specified by a DITA map. The DocBook book can then be processed with the DocBook transform tools. Executed by the docbookComposeDemo target in the run-ant demo build.
docbookDitaref2docbook	Populate a DocBook book by reference to a DITA map. The DocBook book can then be processed with the DocBook transform tools. Executed by the docbookDitarefDemo target in the run-ant demo build.