

**An OASIS White Paper**

## **DITA 1.2 Feature Overview: Conref Push**

Kristen James Eberlein  
Eberlein Consulting

26 October 2009



OASIS (Organization for the Advancement of Structured Information Standards) is a not-for-profit, international consortium that drives the development, convergence, and adoption of e-business standards. Members themselves set the OASIS technical agenda, using a lightweight, open process expressly designed to promote industry consensus and unite disparate efforts. The consortium produces open standards for Web services, security, e-business, and standardization efforts in the public sector and for application-specific markets. OASIS was founded in 1993. More information can be found on the OASIS website at <http://www.oasis-open.org>.

The OASIS DITA Adoption Technical Committee members collaborate to provide expertise and resources to educate the marketplace on the value of the DITA OASIS standard. By raising awareness of the benefits offered by DITA, the DITA Adoption Technical Committee expects the demand for, and availability of, DITA conforming products and services to increase, resulting in a greater choice of tools and platforms and an expanded DITA community of users, suppliers, and consultants.

**DISCLAIMER: All examples presented in this article were produced using one or more tools chosen at the author's discretion and in no way reflect endorsement of the tools by the OASIS DITA Adoption Technical Committee.**

# Table of contents

Conref push .....	4
Overview.....	4
Example of pushing content after an element in a topic .....	4
Adding the @id attribute .....	6
Creating the content to be pushed.....	6
Modifying the DITA map .....	7
Example of pushing content before an element in a topic .....	7
Example of replacing an element in a topic.....	9
Use cases for conref push.....	10
Adding content to auto-generated DITA topics.....	10
Providing a temporary fix for a documentation issue.....	10
Reusing topics between documentation teams .....	10
Modifying third-party documentation.....	10
Rules and restrictions .....	10

## Conref push

DITA 1.2 introduces the conref push feature, which enables users to insert or replace content in DITA topics and maps. This is accomplished by the introduction of a new @conaction attribute with four values: pushreplace, pushbefore, pushafter, and mark.

## Overview

The conref (content reference) mechanism is a core feature of the Darwin Information Typing Architecture (DITA). An elegant transclusion device, the conref mechanism enables significant reuse of DITA elements, topics, and maps, thus reducing translation costs and increasing consistency.

Until DITA 1.2, the conref mechanism only used a pull feature, that is, you could only pull content into DITA topics and maps. With DITA 1.2, you now have the ability to push content into DITA topics and maps, provided that the topics and maps contain elements with @id attributes that identify the places where the content is to be pushed.

This article contains screen captures of DITA source files (as viewed in <oXygen/> XML Editor 10.0) and output generated by the DITA Open Toolkit, version 1.5, milestone 17. This article does not include examples of pushing content into DITA maps. While that functionality is part of the DITA 1.2 standard, it has not yet been implemented in a milestone build of the DITA Open Toolkit.

## Example of pushing content after an element in a topic

Imagine that you are updating a version of the sample files that ship with the DITA Open Toolkit. This information set contains a helpful topic about changing oil in an automobile.

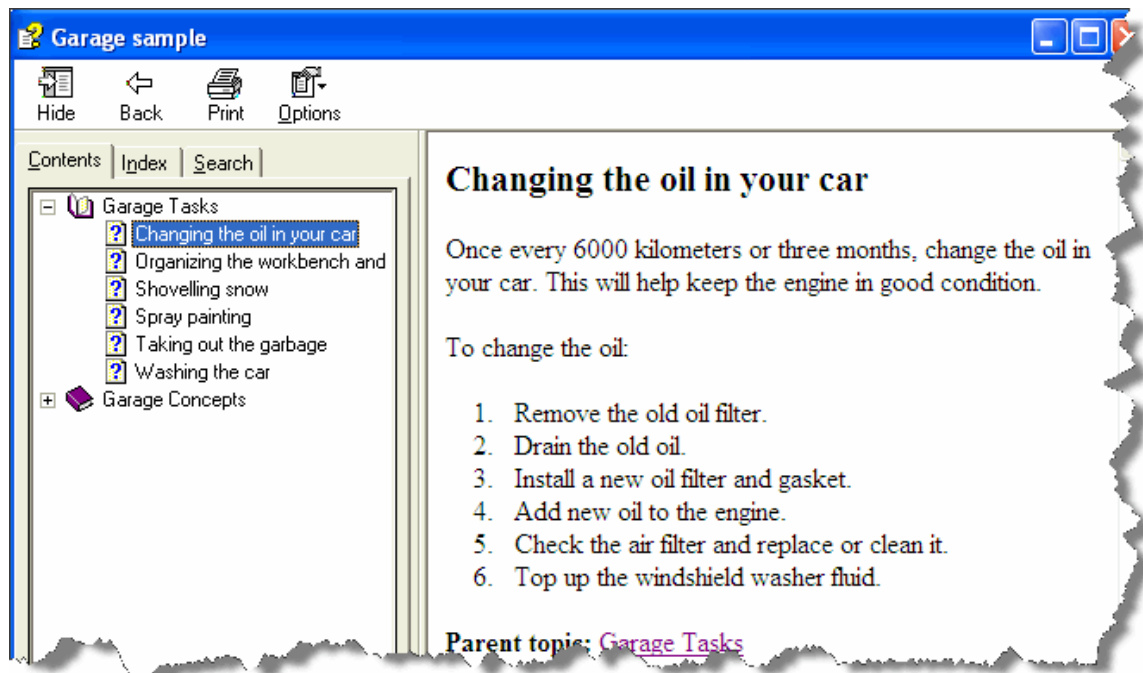


Figure 1. "Changing the oil" topic: HTML Help output

However, the topic fails to address a critical legal issue in your locality: Used motor oil must be properly disposed of. With DITA 1.2, you can push a new <step> element into the "Changing the oil" topic, so that you can generate the following HTML output:

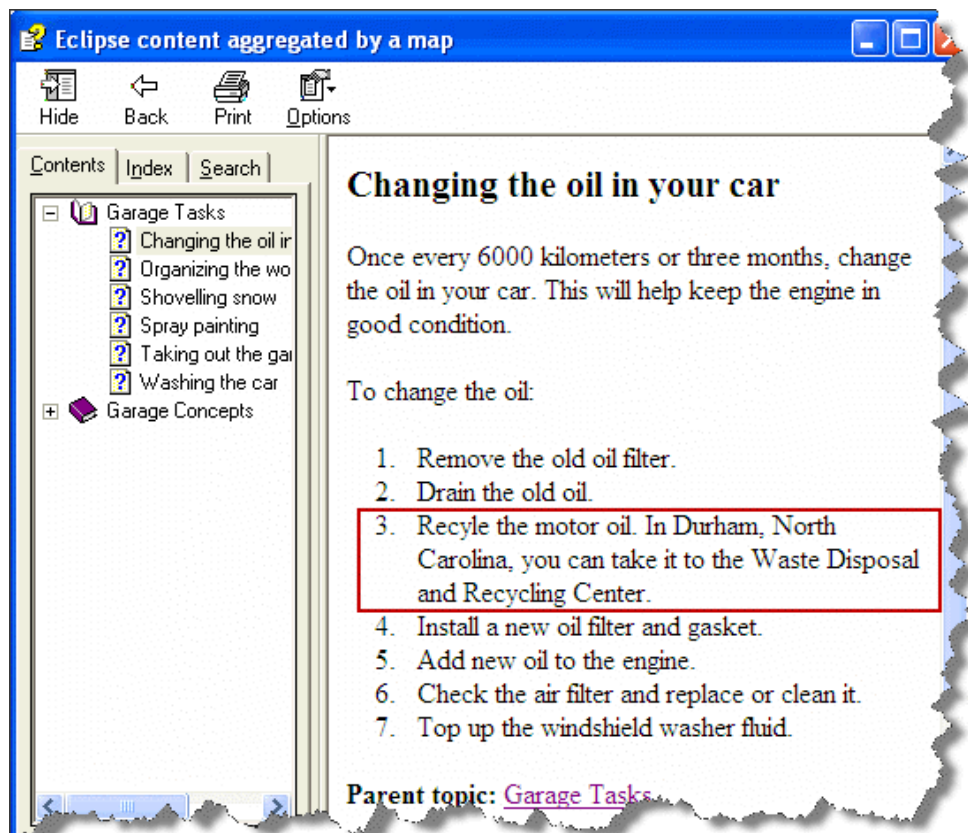


Figure 2. "Changing the oil" topic: HTML Help output after a conref push operation

How was this done? What was required?

The person working with the DITA source files only needed to make the following changes to the information set:

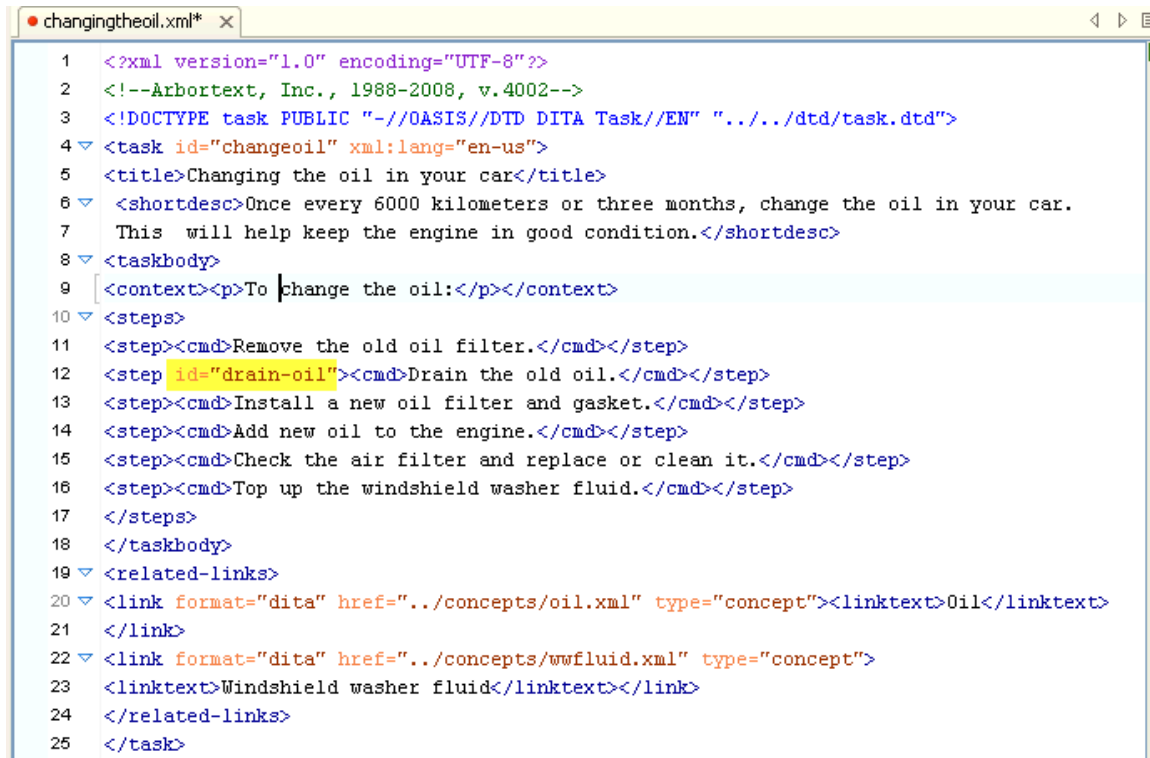
1. Ensure that the original "Changing the oil" topic had an @id attribute for the <step> element that contained the instruction to "Drain the old oil." If she had access to that file, she could add an @id attribute; otherwise, she'd need to request that the person who owns the file add the necessary attribute.
2. Create the <step> element that contains the content to be pushed into the "Changing the oil" topic.

This content either could be added to an existing *conref topic* or created in a new topic. A *conref topic* stores DITA elements that are addressed by the @conref attribute and designed to be used in more than one location. A *conref topic* also is often called a reuse or resource topic; it must be a valid DITA document.

3. Modify the DITA map to reference the *conref topic*, if it does not already include it.

## Adding the @id attribute

The writer arranges for an @id attribute to be added to the applicable <step> element in the "Changing the oil" topic. The modified topic now looks as follows; note the @id attribute that is highlighted in yellow:



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--Arbortext, Inc., 1988-2008, v.4002-->
3 <!DOCTYPE task PUBLIC "-//OASIS//DTD DITA Task//EN" "../dtd/task.dtd">
4 <task id="changeoil" xml:lang="en-us">
5 <title>Changing the oil in your car</title>
6 <shortdesc>Once every 6000 kilometers or three months, change the oil in your car.
7 This will help keep the engine in good condition.</shortdesc>
8 <taskbody>
9 <context><p>To change the oil:</p></context>
10 <steps>
11 <step><cmd>Remove the old oil filter.</cmd></step>
12 <step id="drain-oil"><cmd>Drain the old oil.</cmd></step>
13 <step><cmd>Install a new oil filter and gasket.</cmd></step>
14 <step><cmd>Add new oil to the engine.</cmd></step>
15 <step><cmd>Check the air filter and replace or clean it.</cmd></step>
16 <step><cmd>Top up the windshield washer fluid.</cmd></step>
17 </steps>
18 </taskbody>
19 <related-links>
20 <link format="dita" href="../concepts/oil.xml" type="concept"><linktext>Oil</linktext>
21 </link>
22 <link format="dita" href="../concepts/wwfluid.xml" type="concept">
23 <linktext>Windshield washer fluid</linktext></link>
24 </related-links>
25 </task>
```

Figure 3. "Changing the oil" topic with @id attribute added

## Creating the content to be pushed

The writer now creates a DITA topic to hold the content to be pushed: conref-file.dita. (She also could use an existing conref topic, if one existed.)



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE task PUBLIC "-//OASIS//DTD DITA Task//EN" "http://docs.oasis-open.org/dita/v1.1
3 <task id="conref-file">
4 <title>Conref file</title>
5 <taskbody>
6 <steps>
7 <step conref="changingtheoil.xml#changeoil/drain-oil" conaction="mark">
8 <cmd/>
9 </step>
10 <step conaction="pushafter">
11 <cmd>Recycle the motor oil. In Durham, North Carolina, you can take it to
12 Disposal and Recycling Center.</cmd>
13 </step>
14 </steps>
15 </taskbody>
16 </task>
```

Figure 4. Topic that contains the content to be pushed after

Here is an explanation of the content of the topic:

**The first <step> element** identifies the <step> element after which the content is to be pushed. This happens in two parts:

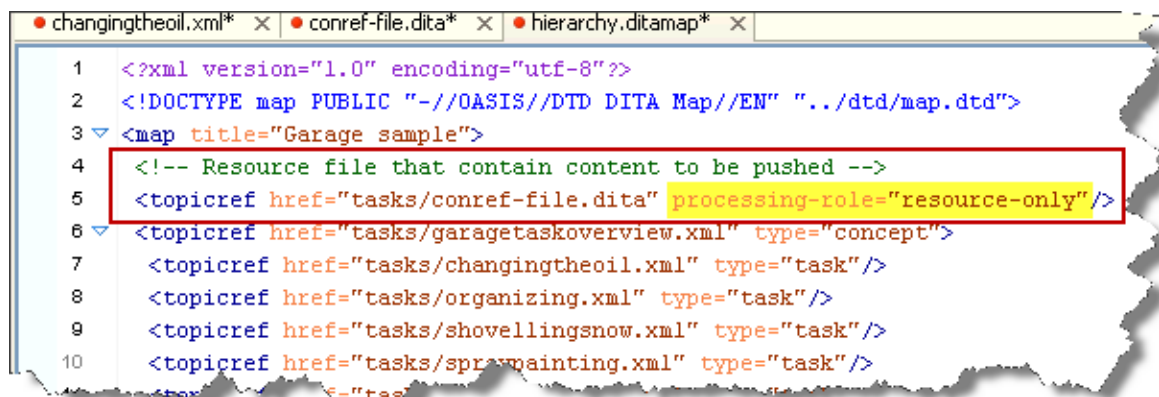
- The @conref attribute uses standard syntax to identify the <step> element.
- The @conaction attribute (new with DITA 1.2) is set to "mark". Setting the @conaction attribute to "mark" indicates that a conref push operation is indicated.

**Note** that the <cmd> element is empty; the content model for the <step> element requires that a <cmd> element be present.

**The second <step> element** specifies the content that is to be pushed into the "Changing the oil" topic. The <cmd> element contains the content, and the @conaction attribute indicates that the <step> element is to be pushed *after* the element with a @conaction attribute set to "mark".

## Modifying the DITA map

The writer then modifies the DITA map for the deliverable so that it references the conref topic that contains the content to be pushed:



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!DOCTYPE map PUBLIC "-//OASIS//DTD DITA Map//EN" "../dtd/map.dtd">
3 <map title="Garage sample">
4 <!-- Resource file that contain content to be pushed -->
5 <topicref href="tasks/conref-file.dita" processing-role="resource-only" />
6 <topicref href="tasks/garagetaskoverview.xml" type="concept">
7 <topicref href="tasks/changingtheoil.xml" type="task"/>
8 <topicref href="tasks/organizing.xml" type="task"/>
9 <topicref href="tasks/shovellingsnow.xml" type="task"/>
10 <topicref href="tasks/spraypainting.xml" type="task"/>
```

Figure 5. Modified DITA map that references the conref topic

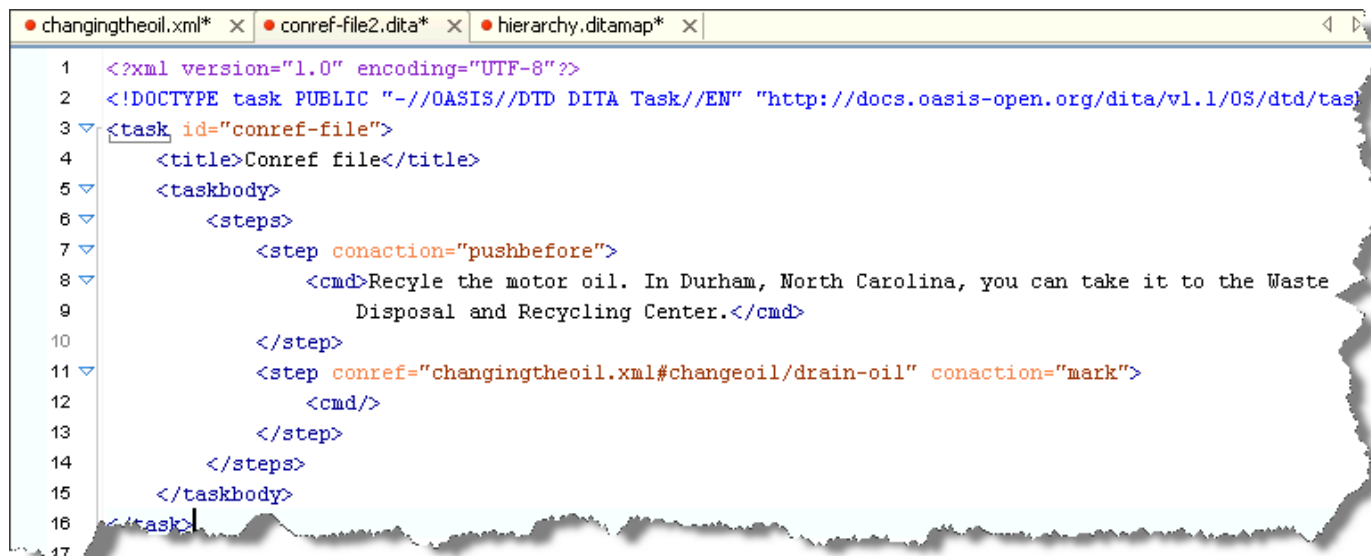
The conref-file.dita topic is referenced in the DITA map by a <topicref> element. Note that the @processing-role attribute (new with DITA 1.2) is set to "resource-only". This attribute indicates that the topic referenced should be used only to resolve conref, keyref, or xref operations; it should not appear in the table of contents (TOC), be found in a search operation, or be linked to.

Note also that the <topicref> element appears at the top of the DITA map and is marked with a comment. This is not necessary – the <topicref> element could be placed anywhere in the DITA map – but reflects best practices and helps to make the DITA map easy for humans to read and scan. If a DITA map contains many <topicref> elements that reference conref or keyref resources, consider building a separate DITA map to contain those references.

## Example of pushing content before an element in a topic

The writer also could push the <step> element *before* the <step> element with the @id attribute. (Let's ignore for the moment that this doesn't make sense, as one would not recycle the used motor oil before draining it from the automobile.) All this would require is a modification to the conref-file.dita topic; the changes that previously were made to the "Changing the oil" and the DITA map would work here, too.

Here is what the conref topic would look like, if the writer wanted to push the new <step> element *before* the <step> element with the @id attribute:



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE task PUBLIC "-//OASIS//DTD DITA Task//EN" "http://docs.oasis-open.org/dita/v1.1/OS/dtd/task.dtd">
3 <task id="conref-file">
4   <title>Conref file</title>
5   <taskbody>
6     <steps>
7       <step conaction="pushbefore">
8         <cmd>Recycle the motor oil. In Durham, North Carolina, you can take it to the Waste
9           Disposal and Recycling Center.</cmd>
10      </step>
11      <step conref="changingtheoil.xml#changeoil/drain-oil" conaction="mark">
12        <cmd/>
13      </step>
14    </steps>
15  </taskbody>
16 </task>
```

Figure 6. Topic that contains the content to be pushed before

**Note** that the <step> element that contains the content to be pushed is placed *before* the <step> element with the conaction="mark" attribute; it also has a @conaction attribute that is set to "pushbefore".

When the DITA map that references the modified conref-file.dita topic is processed, the following output is generated:

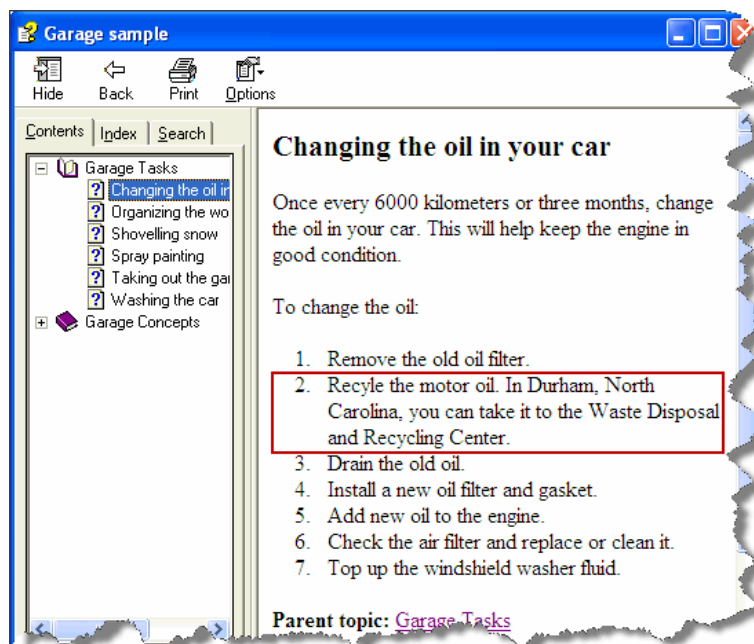


Figure 7. "Changing the oil" topic: HTML Help output after a push before operation

## Example of replacing an element in a topic

The writer also could simply choose to replace the <step> element in the "Changing the oil" topic. Here is what the conref topic would contain:

A screenshot of a text editor window titled 'conref-file.dita'. The editor shows XML code for a DITA task. The code is as follows:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE task PUBLIC "-//OASIS//DTD DITA Task//EN" "http://docs.oasis-open.org/dita/v1.1/OS/dtd/task.
3 <task id="conref-file">
4   <title>Conref file</title>
5   <prolog/>
6   <taskbody>
7     <steps>
8       <step conref="changingtheoil.xml#changeoil/drain-oil" conaction="pushreplace">
9         <cmd>Drain the motor oil.</cmd>
10        <info>Be sure to follow local regulations concerning the disposal of used
11        motor oil.</info>
12      </step>
13    </steps>
14  </taskbody>
15 </task>
```

The XML element on line 8 is highlighted with a red rectangular box.

Figure 8. Topic that contains the content to be used as a replacement

**Note** that for a conref pushreplace operation, only one element is needed. The @conref attribute specifies the element to be replaced, and the @conaction attribute is set to "pushreplace".

Here is a screen shot of the HTML Help that is generated by the DITA Open Toolkit:

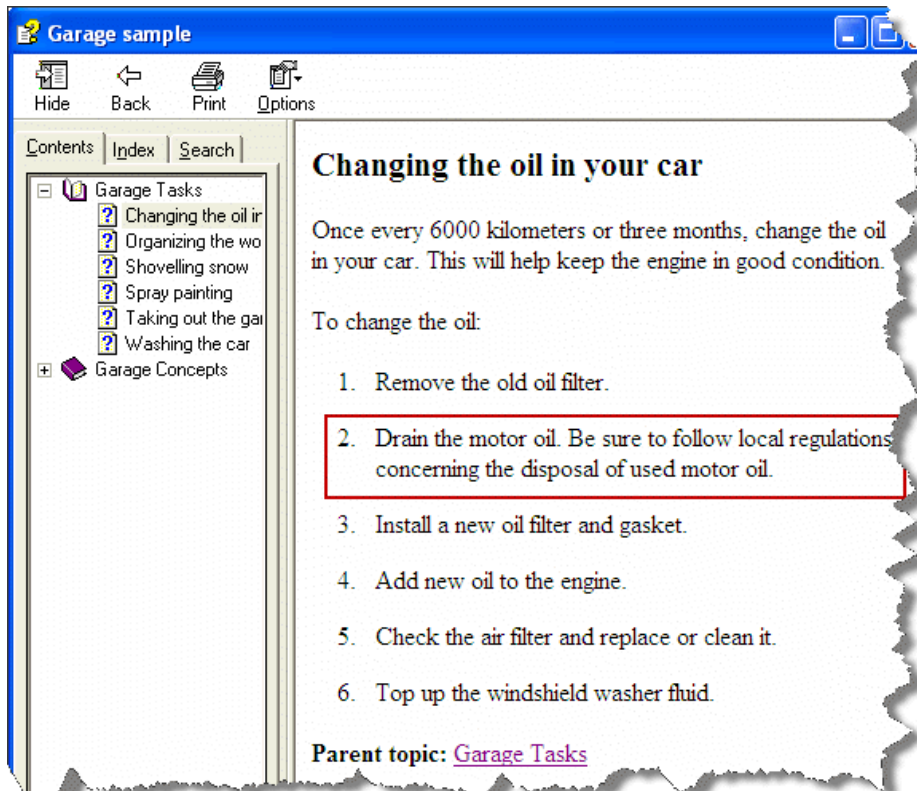


Figure 9. "Changing the oil" topic: HTML Help after a push replace operation

## Use cases for conref push

Here are some scenarios in which companies use the conref push mechanism.

### ***Adding content to auto-generated DITA topics***

A software company generates hundreds of API reference topics every night, using an automated process that builds the topics from the product source code. The company's technical writer wants to add some additional content to the files, to make them more useful for end readers, but he knows that his edits would be overwritten every evening.

The conref push mechanism provides a solution. The writer drafts the content to be pushed into the API reference topics by referencing persistent @id attributes in the auto-generated topics. Every night the topics are regenerated, and every night the human-authored text is pushed into the topics.

### ***Providing a temporary fix for a documentation issue***

A printer manufacturer discovers that Microsoft has changed a configuration setting in Windows; this change invalidates the manufacturer's instructions for installing a printer driver. They need to issue a patch to the installation topic on their Web site. Since the company will have an updated driver in a matter of days, they do not want to change the source for the installation topic. Using the conref push mechanism, they can create a temporary fix that will be simple to remove once the printer driver is updated.

### ***Reusing topics between documentation teams***

A large company uses DITA for their technical documentation. One product, an integrated development environment (IDE), is the basis for another software product. The installation process for the two applications is almost (but not quite) identical; there is one extra window for the second product. The IDE is released before all the development work on the second product is completed. However, since the information developers for the IDE know that another documentation team will reuse their installation topic, they added @id attributes in the necessary places. The technical writers for the second product then use conref push to add the content for the additional installation step.

### ***Modifying third-party documentation***

A hardware company provides their customers with a documentation package that includes some third-party documentation. They need to add a warning to the third-party documentation, but for contractual reasons, they do not want to ask the third party to make the change. Since all the documentation is sourced in DITA – and the third-party documentation contains @id attributes in the necessary places, the hardware company can use conref push to add the warning.

## Rules and restrictions

All the rules and restrictions are designed to ensure that the result of the push operation is valid DITA. This section uses the following terminology:

The *referenced element* is the element addressed by the @conref attribute; it is the element that is replaced, preceded, or followed by the content that is pushed.

The *referencing element* is the element with the @conref attribute.

- In the case of a push and replace operation, the referencing element contains the content that is pushed.

- In the case of a push before or a push after operation, the referencing element specifies the location where content is pushed; a second element contains the content that is pushed.

For all push operations, the referencing element must be of the same type (or more specialized) than the referenced element. For example, both of the following scenarios are valid scenarios:

- A <step> element is the referencing element, and a <step> element is the referenced element. This is what is demonstrated in the examples in this feature article.
- A <step> element is the referencing element, and a <li> element is the referenced element. This is possible because the <step> element is specialized from the <li> element. During processing, the <step> element is generalized back to a <li> element.

However, the referencing element **cannot** be more generalized than the referenced element. For example, one cannot push a <li> element into a <step> element. The content model for a <li> element permits many children elements (for example, <p> elements) that would not be valid within the <step> element.

For push before or push after operations, some additional restrictions apply:

- The element that specifies the conaction="mark" attribute must be of the same type as the element that specifies the push operation (either conaction="pushbefore" or conaction="pushafter"). Both elements will appear in the output.
- The parent of the referencing element must be of the same type as (or more specialized than) the parent of the referenced element. This ensures the validity of the target; for example, comparing the parent elements will prevent a <title> element that is a child of <section> from being pushed before (or after) a <title> element that is a child of <figure>.

What happens if multiple push operations target the same element?

It is an error for two or more elements to attempt to replace the same referenced element. According to the DITA 1.2 specification: "Applications may, but need not, warn users if more than one element attempts to replace a single target."

If two or more elements attempt to push content before the referenced element, the order in which the pushed content is processed is undefined.

If two or more elements attempt to push content after the same referenced element, the order in which the pushed content is processed is undefined.