

**An OASIS White Paper**

## **DITA 1.2 Referencing a Range of Elements: Feature Description**

Joe Gelb, Suite Solutions

21 September, 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

Table of Contents.....	3
Overview .....	4
Purpose.....	4
Usage .....	4
Restrictions.....	4
Examples .....	4
Example 1: Conref in a topic.....	4
Example 2: Conref in a map .....	6
Example 3: Conref to a range where siblings are different element types.....	8

## Overview

A content reference (conref) can be made to a range of adjacent (sibling) elements. The `conref` attribute specifies the first element in the range to be referenced and the new `conrefend` attribute specifies the last element in the range.

## Purpose

With DITA 1.1, the content reference (conref) feature allows authors to reference a single element for each conref reference. For example, if an author wishes to reference a series of steps in a task, either the entire `<steps>` element would need to be referenced (which would include all the child `<step>` elements) or a separate conref would need to be used to reference each individual `<step>`. DITA 1.2 introduces the ability to reference a range of adjacent elements (siblings) by specifying the start and end element.

## Usage

To make a content reference (conref) to a range of elements:

1. Specify the first element of the target range using the `conref` attribute.
  2. Specify the last element of the target range using the new `conrefend` attribute
- The same addressing rules that apply to `conref` apply to `conrefend`. Note that only the first and last elements in the range need to have an id. There is no need to assign an id to the other siblings in the range.

Conref ranges can be used in topics or maps.

## Restrictions

Be aware of the following guidelines and restrictions when using conref ranges:

1. The element specified in the `conref` attribute as starting the range must come before the element specified as the `conrefend` in the target file.
2. Both the first and last elements in the range must be siblings; that is, they must be children of the same parent element.
3. The parent of the element with the `conref` should be the same element type (or be specialized from the same element type) as the parent of the elements being referenced. This constraint was put in place to minimize the potential for invalid content after the reference is resolved. In practice, it is up to the processing engine to determine how to handle these cases.
4. Both the first and last elements in the range should be the same element type (or be specialized from the same element type). For example, an author can reference a range of paragraphs (`<p>` elements), but may not start with a `<p>` and end with a `<table>`. However, an author can reference a range which would start with a `<p>`, be followed by other body elements (such as tables, lists, images), and end with another `<p>`. This constraint was put in place to minimize the potential for invalid content after the reference is resolved. In practice, it is up to the processing engine to determine how to handle these cases.

## Examples

### Example 1: Conref in a topic

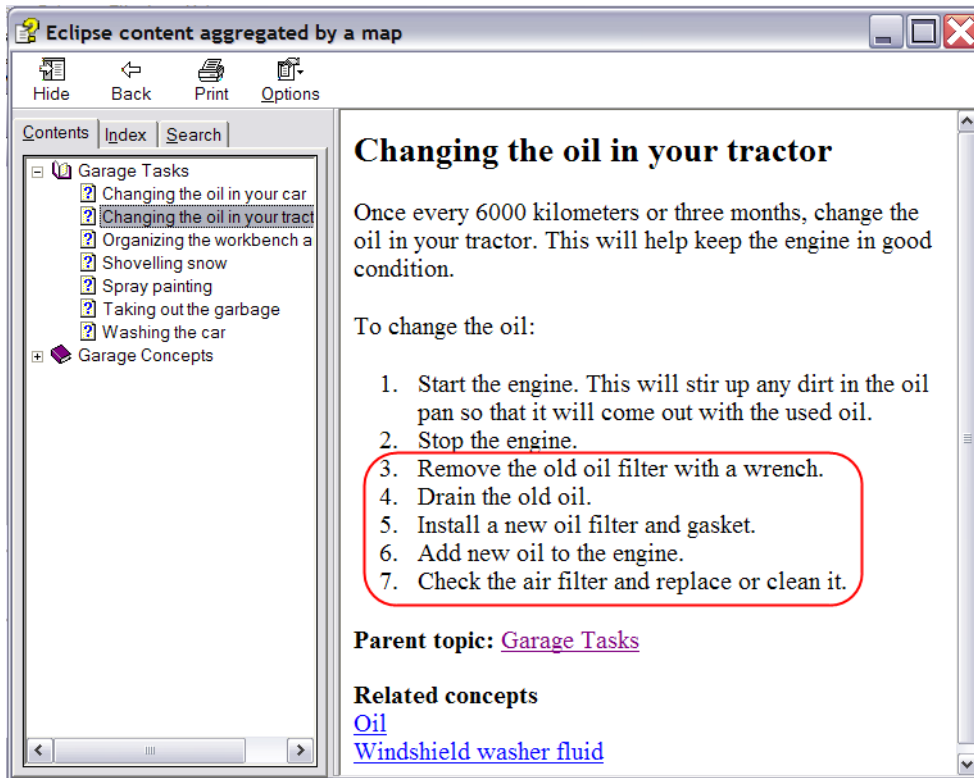
In this example, steps from the topic `changingtheoil.xml` will be referenced in the topic `changingtheoil_tractor.xml`.

### changingtheoil.xml

```
<task id="changeoil" xml:lang="en-us">
<title>Changing the oil in your car</title>
...
<taskbody>
...
  <steps>
    <!-- the first and last step elements to be referenced
          have id's -->
    <step id="step_removeoldfilter"><cmd>Remove the old oil filter
with a wrench.</cmd></step>
    <step><cmd>Drain the old oil.</cmd></step>
    <step><cmd>Install a new oil filter and gasket.</cmd></step>
    <step><cmd>Add new oil to the engine.</cmd></step>
    <step id="step_checkfilter"><cmd>Check the air filter and replace
or clean it.</cmd></step>
    <step><cmd>Top up the windshield washer fluid.</cmd></step>
  </steps>
</taskbody>
...
</task>
```

### changingtheoil\_tractor.xml

```
<task id="changeoil_tractor" xml:lang="en-us">
<title>Changing the oil in your tractor</title>
...
<taskbody>
  <steps>
    <!-- the first 2 steps are followed by a reference
          to a range of steps in changingtheoil.xml -->
    <step><cmd>Start the engine. This will stir up any dirt
in the oil pan so that it will come out with the used
oil.</cmd></step>
    <step><cmd>Stop the engine.</cmd></step>
    <step conref="changingtheoil.xml#changeoil/step_removeoldfilter"
conrefend="changingtheoil.xml#changeoil/step_checkfilter"><cmd>
</cmd></step>
  </steps>
</taskbody>
...
</task>
```



*Result of reference: Five consecutive steps are included*

## Example 2: Conref in a map

In this example, a map was created to publish only those topics related to motor vehicles. Two ranges of <topicref> elements from the map hierarchy.ditamap are referenced in the map hierarchy\_car.ditamap.

Note in this example that the addressing used for targets in another map is different than the addressing used in the last example for targets in another topic. Use the following addressing for targeting elements inside a map:

Target in the same map:

```
conref="#elementID"
```

Target in a different file:

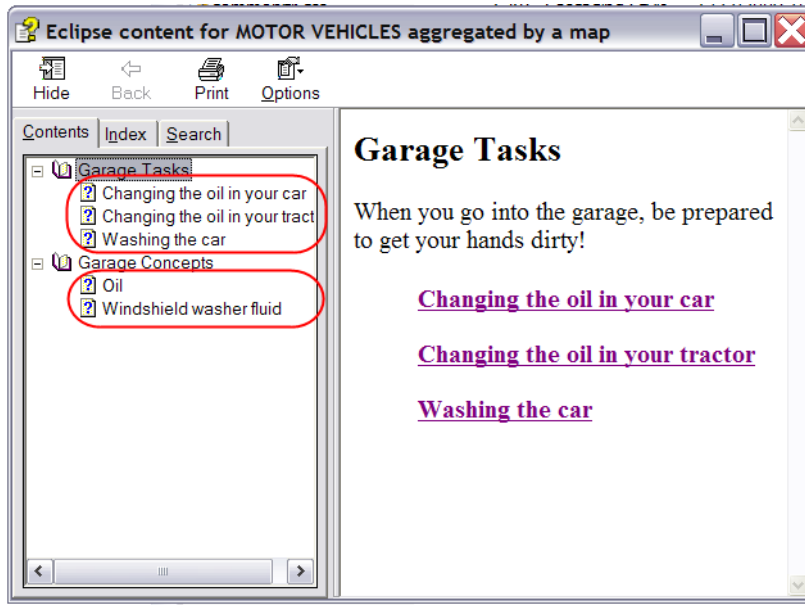
```
conref="othermap.ditamap#elementID"
```

### hierarchy.ditamap

```
<map title="Eclipse content aggregated by a map">
  <topicref href="tasks/garagetaskoverview.xml" type="concept">
    <!-- the first and last topicref elements to be referenced
         have id's -->
    <topicref id="changingoil" href="tasks/changingtheoil.xml"
      type="task"/>
    <topicref href="tasks/changingtheoil_tractor.xml" type="task"/>
    <topicref id="washingcar" href="tasks/washingthecar.xml"
      type="task"/>
    <topicref href="tasks/organizing.xml" type="task"/>
    <topicref href="tasks/shovellingsnow.xml" type="task"/>
    <topicref href="tasks/spraypainting.xml" type="task"/>
    <topicref href="tasks/takinggarbage.xml" type="task"/>
  </topicref>
  <!-- the first and last topicref elements to be referenced
       have id's -->
  <topicref href="concepts/garageconceptsoverview.xml" type="concept">
    <topicref id="oil" href="concepts/oil.xml" type="concept"/>
    <topicref id="wwfluid" href="concepts/wwfluid.xml"
      type="concept"></topicref>
    <topicref href="concepts/lawnmower.xml" type="concept"/>
    <topicref href="concepts/paint.xml" type="concept"/>
    <topicref href="concepts/shelving.xml" type="concept"/>
    <topicref href="concepts/snowshovel.xml" type="concept"/>
    <topicref href="concepts/toolbox.xml" type="concept"/>
    <topicref href="concepts/tools.xml" type="concept"/>
    <topicref href="concepts/waterhose.xml" type="concept"/>
    <topicref href="concepts/wheelbarrow.xml" type="concept"/>
    <topicref href="concepts/workbench.xml" type="concept"/>
  </topicref>
</map>
```

### hierarchy\_car.ditamap

```
<map title="Eclipse content for MOTOR VEHICLES aggregated by a map">
  <topicref href="tasks/garagetaskoverview.xml" type="concept">
    <!-- this topicref references a range of topicrefs from another
         map -->
    <topicref conref="hierarchy.ditamap#changingoil"
      conrefend="hierarchy.ditamap#washingcar" />
  </topicref>
  <topicref href="concepts/garageconceptsoverview.xml" type="concept">
    <!-- this topicref references a range of topicrefs from another
         map -->
    <topicref conref="hierarchy.ditamap#oil"
      conrefend="hierarchy.ditamap#wwfluid" />
  </topicref>
</map>
```



*Result of reference: Two ranges of topicrefs are included*

### Example 3: Conref to a range where siblings are different element types

This example demonstrates how the range can include a variety of content, as long as the starting and ending elements share the same parent, and are themselves the same element type as (or a specialization of) that of the referencing element. Here, the range being referenced starts and ends with <p> elements, but also includes a sibling of a different type. That sibling also has child elements.

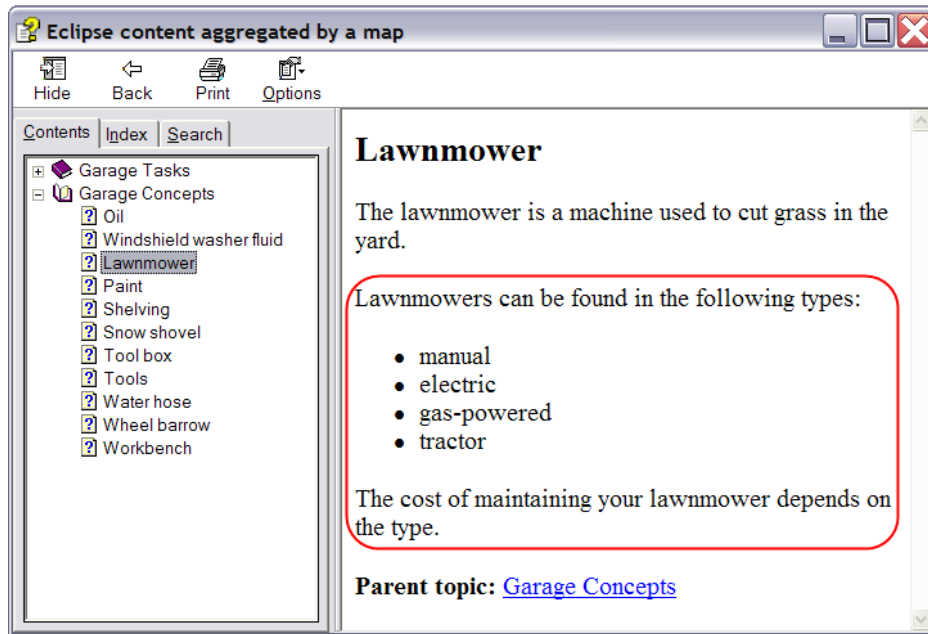
#### lawnmower\_types.xml

```
<concept id="lawnmowertypes" xml:lang="en-us">
<title>Types of Lawnmowers</title>
<conbody>
  <p id="types">Lawnmowers can be found in the following types:</p>
  <ul id="typelist">
    <li>manual</li>
    <li>electric</li>
    <li>gas-powered</li>
    <li>tractor</li>
  </ul>
  <p id="maint">The cost of maintaining your lawnmower depends on the
  type.</p>
</conbody>
</concept>
```

#### lawnmower.ditamap

```
<concept id="lawnmowerconcept" xml:lang="en-us">
<title>Lawnmower</title>
<conbody>
  <p>The lawnmower is a machine used to cut grass in the yard.</p>
  <!-- the following is a reference to a range of elements -->
  <p conref="lawnmower_types.xml#lawnmowertypes/types"
  conrefend="lawnmower_types.xml#lawnmowertypes/maint" />
</conbody>
</concept>
```





*Result of reference: Range of different content is included*