

An OASIS DITA Adoption Technical Committee Publication

DITA 1.2 Feature Article: Task Model

JoAnn Hackos
On behalf of the OASIS DITA Adoption Technical Committee

Date: July 9 2010



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.

This white paper was produced and approved by the OASIS DITA Adoption Technical Committee as a Committee Draft. It has not been reviewed and/or approved by the OASIS membership at-large.

Copyright © 2010 OASIS. All rights reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English. The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Document History

Revision	Date	Author	Summary

Table of Contents

Task Model Feature Article	7
Purpose of the General Task Model	7
Precautions in mixing task models	7
Steps-informal	7
Steps and steps-unordered	8
General Task Model example	8
Steps-unordered example	9
Steps-informal example	9



Task Model Feature Article

DITA 1.2 introduces the General Task Model as an alternative to the DITA 1.0 and 1.1 Strict Task Model. It also adds capabilities to the <steps> content unit for both task models.

Use the DITA General Task Model when your information model calls for a task structure that is not well supported by the DITA Strict Task Model or when you need to constrain the elements in a task in a way that is unique to your organization's needs. Use the DITA General Task Model to convert existing tasks to DITA that do not yet conform to your task information model.

DITA 1.2 also includes the new Machinery Task Model, designed to conform to the requirements of the machine industry, especially in the construction of pre- and post-requirements. The Machinery Task Model is described in another features article.

Purpose of the General Task Model

The DITA Strict Task Model is the most restrictive of the DITA information types. For example, the DITA Strict Task Model, originally introduced in DITA 1.0 and maintained in DITA 1.1 and 1.2, requires that pre-requisites precede context in the <conbody> and does not permit titles for pre-requisites, context, result, or post-requisite content units.

The new DITA General Task Model, introduced with DITA 1.2, is less restrictive to support variations in the way that organizations create procedural information. The General Task Model can be the basis for specific constraints required by the organization or it can be used to support a more opened way of developing task information.

The new DITA General Task Model allows the following:

- Add a <section> with an optional <title> before the steps in the task.
- Vary the order of <prereq>, <context>, and <section> before the steps in the task. It's common for some organizations to present <context> information before <prereq>, and, in the past, the DITA-OT stylesheets had to be customized to switch them around, which is beyond the capability of most users who are not well-versed in advanced XSLT coding techniques.
- Select the new <steps-informal> content unit, which allows you to construct steps using ordered and unordered lists, paragraphs, and other less semantically specified elements than <step> and <cmd>.
- Include more than one example or pre-requisite in a task after the completion of the steps.

Precautions in mixing task models

Note that the DITA General Task Model, the DITA Strict Task Model, and the Machinery Task Model are not mutually compatible. Both DITA Strict Task Model and the Machinery Task Model have been redesigned as constraints of the General Task Model.

You are not able to use a conref between a constrained and a more general task model. A constrained task cannot conref from an unconstrained task. Conref requires that the conref target be at least as constrained (and compatibly constrained) as the conref source (that is, the document making the conref link). Otherwise, you cannot be assured that the conrefed data is valid in the using context.

We advise organizations to use only one task model unless the content in two areas of publications is completely mutually exclusive and will never be shared. Note, however, that a fundamental DITA principle is the ability to exchange content throughout an organization. It is, therefore, best to avoid using more than one task model.

Steps-informal

The <steps-informal> element allows you to describe procedural task information that would not normally be ordered as steps, such as a group of generic procedures that may all be applied in a particular situation. A typical example



might be a Troubleshooting procedure that identifies one symptom and then provides more than one procedure for correcting the problem. This element is also designed to be used for specialization of the structure of steps.

The `<steps-informal>` element can contain `` or `` elements, which are less strictly structured than `<step>` elements and therefore offer the following benefits:

- The contents of an `` can often be copied directly from HTML and pasted into DITA; this is not the case for `<steps>` or `<steps-unordered>` elements.
- When converting legacy content, it may be easier to convert numbered lists to `` elements than to `<steps>` elements

One drawback of `<steps-informal>` is that it cannot contain a `<stepsection>` element, as described below.

Steps and steps-unordered

Both the General Task Model and the Strict Task Model incorporate new elements into the `<steps>` and `<steps-unordered>` content units.

Within the `<steps>` or `<steps-unordered>` elements, you can add expository text using the new optional `<stepsection>` element. The optional `<stepsection>` may precede any `<step>`, allowing you to provide information that is not actually part of a step. Note that you may have to adjust your stylesheet to ensure that the `<stepsection>` is not numbered with the steps, if that is the behavior you prefer.

You can also add one or more optional notes or hazard statements before the `<cmd>` element in a step.

If you include the Hazard Domain with your DITA shell model, you can insert the new `<hazardstatement>` before a command. Many DITA users have identified the need for warnings, cautions, or other hazards to precede steps but should not be written as a part of the previous step. The new placement of notes and hazard statements helps to consolidate information around a step and simplifies the reuse of step content.

General Task Model example

In this example using the General Task Model, the `<context>` element precedes the `<prereq>` element. In addition, a `<section>` is added with a `<title>` before the steps begin. The `<steps>` contain a `<stepsection>` before the `<step>` list begins. A warning and a danger occur within `<note>` elements before the `<cmd>` of the step.



Note that additions to the `<steps>` appears in both the General Task Model and the Strict Task Model.

```
<task>
<title>Lifting the pump assemblies</title>
<shortdescrip>Use this procedure to lift the PS ANSI combo pump units.</shortdescrip>

<taskbody>
<context>PS units may be mounted with equipment installed or with no equipment
installed.</context>
<prereq> PS units are transported on wooden pallets via forklift truck
to the area where they will be installed.
<note othertype="warning" type="other">Never transport a PS unit
over a long distance or over rough terrain while it is
suspended from slings.</note></prereq>
<section>
<title>Trained personnel</title>
<p>Ensure that lifting is handled only by trained personnel.</p>
</section>
<steps>
<stepsection> If you are lifting a unit with the motor installed,
perform the first step, otherwise continue with the second step.
</stepsection>
<step><cmd>Check to see that the pump suction nozzle does not interfere
with the lifting sling. If it interferes, remove it
before proceeding.</cmd></step>
<step><cmd>Remove the metal shipping straps that hold the
PS unit to the wooden pallet.</cmd></step>
<step>
<note type="warning">Do not install eyebolts in the PS thread
```



```

inserts to lift the base. This practice imposes lateral loads
on the inserts that they are not designed to withstand.</note>
<cmd>Slip slings under each end of the PS unit as a harness.</cmd></step>
<step>
<note type="danger">Keep hands and feet out from under the
PS unit during these steps. If slings slip and the unit tips
over, severe personal injury or death may result, as well as
irreparable damage to the PS unit.</note>
<cmd>Lift the PS unit a few inches off the pallet and verify
that it hangs reasonably level and that the slings are not prone
to slip out of position.</cmd></step>
<step><cmd>If the sling appears to be unstable, set the PS unit
back on the pallet and reposition the slings.</cmd></step>
.
.
</taskbody></task>

```

Steps-unordered example

This example include a <section> with a <title> prior to the beginning of the <steps-unordered> content unit.

```

<task>
<title>Locating the pump</title>
<shortdescrip>Consider these issues when you decide where to
locate the pump.</shortdescrip>

<taskbody>
<section><title>Applicable units</title>
<p>Apply these standards to JM or JP units.</p></section>

<steps-unordered>
<step><cmd>Locate the pump as near your liquid source as practical.</cmd></step>
<step><cmd>Allow adequate space for servicing and ventilation.</cmd></step>
...
</steps-unordered>
</taskbody>
</task>

```

Steps-informal example

In the next example, the <context> content unit precedes the <prereq> content unit. <steps-informal> allows paragraphs, ordered lists, unordered lists, and other body elements.

```

<task>
<title>Lifting the pump assemblies</title>
<shortdescrip>Use this procedure to lift the PS ANSI combo pump
units.</shortdescrip>

<taskbody>
<context>PS units may be mounted with equipment installed or with
no equipment installed.</context>

<prereq>
<p>PS units are transported on wooden pallets
via forklift truck to the area where they will be installed.
<note othertype="warning" type="other">Never transport a PS
unit over a long distance or over rough terrain while it is
suspended from slings.</note></p>
<p>Ensure that lifting is handled only by trained personnel.</p>
</prereq>

<steps-informal>
<p> If you are lifting a unit with the motor installed,
perform the first step, otherwise continue with the second step.</p>

<ol><li>Check to see that the pump suction nozzle does
not interfere with the lifting sling. If it interferes,
remove it before proceeding.</li>
<li>Remove the metal shipping straps that hold the PS
unit to the wooden pallet.</li></ol>

<p><note type="warning"> Do not install eyebolts in the
PS thread inserts to lift the base. This practice imposes lateral
loads on the inserts that they are not designed to withstand.</note></p>

```



```

<ol><li>Slip slings under each end of the PS unit as a harness.</li></ol>

<p><note type="danger">Keep hands and feet out from under
the PS unit during these steps. If slings slip and the unit
tips over, severe personal injury or death may result, as well as
irreparable damage to the PS unit.</note></p>

<ol><li>Lift the PS unit a few inches off the pallet and verify
that it hangs reasonably level and that the slings are not prone
to slip out of position.</li>
<li>If the sling appears to be unstable, set the PS unit back
on the pallet and reposition the slings.</li></ol>
...
</taskbody></task>

```

<steps-informal> allows authors to use basic HTML for ordered lists, greatly simplifying the DITA markup in comparison with the more semantically rigorous <steps>.

Another possible application of <steps-informal> allows an author to address a very different model of procedural information. A good example is the classic recipe style in which a number of actions are combined in a single numbered step.

Here's a typical recipe structure.

```

<task><title>Argula parmesan salad</title>

<shortdesc>The vegetable Arugula has a distinctive appealingly
peppery flavor. Use just a touch of sweet balsamic vinegar to
enhance rather than mask it.</shortdesc>

<taskbody>
<prereq><sl><li>1 tbsp. plus 2 tsp. red-wine vinegar</li>
<li>1 tsp balsamic vinegar</li>
...</sl>
</prereq>
<context><p>Serves 8 Time 15 minutes </p></context>
<draft-comment>It would be nice to specialize these data
items.</draft-comment>

<steps-informal>
<ol>
<li>In a bowl, whisk together vinegars, garlic, and olive oil.
Add salt and pepper to taste.</li>
<li>Put arugula in a large bowl and toss with herbs and dressing.
</li>
</ol>
</steps-informal>
</taskbody>
</task>

```

Note that this recipe is quite short. Sometimes a single step in a recipe will have 10-15 actions. This example of <steps-informal> would allow authors to indicate a set of steps, perhaps enabling a specialization of the to <recipe-steps>. It would allow a different structure than the strict task mode's sequence of <cmd> and <info>, better accommodating the multiple actions typically described in a recipe "step."