Task elements

Contents

Task elements

Task topics answer "How do I?" questions, and have a well-defined structure that describes how to complete a procedure to accomplish a specific goal. Use the task topic to describe the steps of a particular task, or to provide an overview of a higher-level task. The task topic includes sections for describing the context, prerequisites, actual steps, expected results, example, and expected next steps for a task. For more details on when to use task and other information types, please refer to the DITA architectural specification.

<task>

The <task> element is the top-level element for a task topic. Tasks are the main building blocks for task-oriented user assistance. They generally provide step-by-step instructions that will enable a user to perform a task. A task answers the question of "how to?" by telling the user precisely what to do and the order in which to do it. Tasks have the same high-level structure as other topics, with a title, short description and body.

Usage information

Note: Beginning with DITA 1.2, the DTD and Schema packages distributed by OASIS contain two task models. The general task model allows two additional elements inside the task body (<section> and <steps-informal>); it also allows multiple instances and varying order for the <prereq>, <context>, and <section> elements. The strict task model maintains the order and cardinality of the DITA 1.0 and 1.1 <taskbody> content model. This strict task is implemented with a constraint module.

See the *taskbody* description for additional details about the two models and for a description of impacts to DITA 1.1 documents.

Specialization hierarchy

The <task> element is specialized from the <topic> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* (with a narrowed definition of @id, given below), *Architectural attribute group*, and *outputclass*.

@id (REQUIRED)	An anchor point. This ID is usually required as part of the @href or @conref syntax when cross referencing or reusing content within the topic; it also enables <topicref> elements in DITA maps to optionally reference a specific topic within a DITA document. This attribute is defined with the XML Data Type ID.

Example

```
<task id="sqlj">
 <title>Creating an SQLJ file</title>
 <taskbody>
  <context>Once you have set up SQLJ, you need to create a new
 SQLJ file.
  </context>
  <steps>
   <step><cmd>Open...</cmd></step>
  </steps>
 </taskbody>
```

```
</task>
```

<taskbody>

The task body contains information specific to completing a task, such as prerequisites, contextual information, and steps.

Disposition: / Status:
Removed this which seems obsolete for 2.0: DITA 1.2 introduced a much looser <taskbody> content model in order to allow for more variations in the structure of a task. A constraint module was also provided in order to maintain compatibility with the previous strict model; this constraint is used in the default task distributed by OASIS.

Usage information

Disposition: / Status:
For DITA 2.0 it seems the information about using 1.1 level markup is obsolete but leaving in as we refactor.

Note: Beginning with DITA 1.2, the DTD and Schema packages distributed by OASIS contain two task models. The general task model allows two additional elements inside the task body (<section> and <steps-informal>); it also allows multiple instances and varying order for the <prereq>, <context>, and <section> elements. The strict task model maintains the order and cardinality of the DITA 1.0 and 1.1 <taskbody> content model. This strict task is implemented with a constraint module.

Authors that use the default task DTD or Schema provided by OASIS will continue to see the strict task model when upgrading to DITA 1.2 or DITA 1.3. Authors wishing to use the general task model will need to migrate their DITA 1.1 documents to reference the general task document type shell.

DITA document type shells that include the task module as-is, or that specialize the <task> element without specializing <taskbody>, will also need to include the strict taskbody constraint module in order to maintain the order and cardinality of prior DITA versions.

Task specializations that specialize the <taskbody> element will not be affected by the new model, although they can be updated as needed to take advantage of the new elements.

Specialization hierarchy

The <taskbody> element is specialized from the <topic> element in the topic module.

Content models

See appendix for information about this element in OASIS document type shells.

Attributes

The following attributes are available on this element: *Universal attribute group* (without the Metadata attribute group), @base from the *Metadata attribute group*, and *outputclass*.

Example

See *task* on page 3.

<prereq>

Prerequisites are things that users must know or tasks that users must perform before starting the current task.

Formatting expectations

Implementations *MAY* render prerequisite links from the related-links section together with the <prereq> content.

Specialization hierarchy

The prereq<> element is specialized from the section element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<task id="sqlj">
 <title>Creating an SQLJ file</title>
 <taskbody>
  <prereq>Before creating a new SQLJ file, you must
  log in to the SQLJ server.</prereq>
 </taskbody>
</task>
```

<context>

The task context provides background information that helps users understand the purpose of the task and what they will gain by completing it.

Usage information

This section should be brief and does not replace or recreate a concept topic on the same subject, although the context section might include some conceptual information.

Specialization hierarchy

The <context> element is specialized from the <section> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<task id="sqlj">
<title>Creating an SQLJ file</title>
<taskbody>
<context>Once you have set up SQLJ, you need to create a
new SQLJ file.</context>
</taskbody>
</task>
```

<steps>

The <steps> element provides the main content of a task topic. The task is described as a series of steps that the user must follow to accomplish the task. At least one <step> element is required inside the <steps> element.

Formatting expectations

Steps with only a single step might be rendered as a paragraph rather than as a list. Two or more steps should typically be rendered as an ordered list. If all of the contained steps are simple (that is, have no more than a <cmd> element each) then the step list should default to compact. Otherwise it should be rendered as expanded (with blank lines between each step).

Specialization hierarchy

The <steps> element is specialized from the <ol> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<task id="sqlj">
<title>Creating an SQLJ file</title>
 <taskbody>
 <context>Once you have set up SQLJ, you need to create a new
 SQLJ file.</context>
  <steps>
   <step>
    <cmd>In a text editor, create a new file.</cmd>
   </step>
   <step>
    <cmd>Enter the first query statement.</cmd>
   </step>
  </steps>
 </taskbody>
</task>
```

<steps-informal>

Informal steps allow authors to describe procedural task information without placing each step in an individual container element, which is a requirement of the related <steps> and <steps-unordered> elements.

Usage information

For example, <steps-informal> might contain a paragraph that describes more than one step in a single sentence, or it might contain sentences that mix steps together with information about the steps.

Specialization hierarchy

The <steps-informal> element is specialized from the <section> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<steps-informal>
 <p>Put the soil in the container any old way. It doesn't really
 matter how
you do it as long as it is at least 12 cm deep. Once the soil is
 in place,
water appropriately and wait.</p>
</steps-informal>
```

<steps-unordered>

Like the <steps> element, the <steps-unordered> element provides the main content of a task topic, but particularly for cases in which the order of steps might vary from one situation to another. At least one <step> element is required inside the <steps-unordered> element.

Formatting expectations

Steps with only a single step might be rendered as a paragraph rather than as a list. Two or more steps should typically be rendered as an unordered list. If all of the contained steps are simple (that is, have no more than a <cmd> element each) then the step list should default to compact. Otherwise it should be rendered as expanded (with blank lines between each step).

Specialization hierarchy

The <steps-unordered> element is specialized from the <ul> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<task id="sqlj">
 <title>Creating an SQLJ file</title>
 <taskbody>
  <context>Once you have set up SQLJ, you need to create a new
 SQLJ file.</context>
  <steps-unordered>
   <step><cmd>In a text editor, create a new file.</cmd></step>
  </steps-unordered>
 </taskbody>
</task>
```

<step>

A step represents an action that a user must follow to accomplish a task.

Usage information

Disposition: / Status:
The following infromation was part of the shortdesc in 1.3. I think all of it should be removed or rephrased:

- *Each step in a task must contain...* We shouldn't be explicit about content models; this tends to get us into trouble as the spec moves forward. Might be enough to just remove "<cmd> element"

- *Beginning with DITA 1.2...* I think this sentence should be considered obsolete when it comes to 2.0.
- *The <step> element can also contain...* This could probably be shortened and moved back into the shortdesc - even combined with the one sentence?

Each step in a task must contain a command <cmd> element which describes the particular action the user must do to accomplish the overall task. Beginning with DITA 1.2, it is possible to place a <note> element before the command in order to notify the user of dangers or other important information about the step. The <step> element can also contain additional optional information about the step, such as sub-steps, a list of choices, or result information.

Specialization hierarchy

The <step> element is specialized from the <li> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* (with a narrowed definition of @importance, given below) and *outputclass*.

importance	Describes whether the current <step> or <substep> is optional or required. Output processors might highlight steps that are optional or required. Allowed values are "optional", "required", or *-dita-use-conref-target*.

Example

```
<task id="sqlj">
<title>Creating an SQLJ file</title>
<taskbody>
<context>Once you have set up SQLJ, you need to create a new
 SQLJ file.</context>
<steps>
<step>
 <cmd>Select <menucascade><uicontrol>File</
uicontrol><uicontrol>New</uicontrol></menucascade>.</cmd>
 <info>New files are created with default values based on a
 standard template.</info>
</step>
</steps>
</taskbody>
</task>
```

<stepsection>

The <stepsection> element provides expository text before a <step> element. Although the element is specialized from <li> and has the same content model as a list item, it is not intended to represent a step in a task.

Formatting expectations

DITA applications which render <stepsection> elements among the <step> elements must provide a way to number the steps without numbering the <stepsection> elements (although this does not need to be the only or default presentation).

Specialization hierarchy

The <stepsection> element is specialized from the <li> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<steps>
  <step><cmd>Get out a bowl</cmd></step>
  <stepsection>The next two steps are very important!</
stepsection>
  <step><cmd>Put on safety gloves</cmd></step>
  <step><cmd>Put on goggles</cmd></step>
  <step><cmd>Pour milk and cereal into the bowl</cmd></step>
</steps>
```

The sample above would typically be rendered with "Get out a bowl" as step number one, "Put on safety gloves" as step number two, and "The next two steps are very important!" as an unnumbered item in between the first two items.

<cmd>

A command provides an instruction for completing a step.

Usage information

A command provides the active voice instruction to the user for completing the step, and should not be more than one sentence. If the step needs additional explanation, place the explanation in an <info> element following the <cmd>.

Specialization hierarchy

The <cmd> element is specialized from the <ph> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group*, *outputclass*, and *@keyref*.

Example

```
<step><cmd>In a text editor, create a new file.</cmd></step>
```

<info>

The <info> element provides additional information about the step.

Specialization hierarchy

The <info> element is specialized from the <itemgroup> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<step><cmd>Type a name for the widget.</cmd>
```

```
<info>The widget name is created when you configure the widget
in the Widget Configuration Dialog. It is not an actual class
name or file name, just a label for the widget as used in this
application.</info>
</step>
```

<substeps>

Sub-steps allow authors to break a step down into a series of separate actions.

Usage information

The <substeps> element should be used only if necessary. Try to describe the steps of a task in a single level of steps. If you need to use more than one level of <substep> nesting, you should probably rewrite the task to simplify it.

Specialization hierarchy

The <substeps> element is specialized from the <ol> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<substeps>
<substep><cmd>Hold pencil in a steady, level position.</cmd></
substep>
<substep><cmd>Turn handle until resistance diminishes.</cmd>
<info>Note: initially, it might be somewhat difficult to turn
 the handle if
pencil has never been sharpened before.</info></substep>
<substep><cmd>To determine if pencil is sharp, remove it from
 the sharpener
and inspect the tip.</cmd></substep>
</substeps>
```

<substep>

A <substep> element has the same structure as a <step>, except that it does not allow lists of choices or sub-steps within it, in order to prevent unlimited nesting of steps.

Specialization hierarchy

The substep<> element is specialized from the <li> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* (with a narrowed definition of @importance, given below) and *outputclass*.

importance	Describes whether the current <step> or <substep> is optional or required. Output processors might highlight

steps that are optional or required. Allowed values are "optional", "required", or *-dita-use-conref-target*.

Example

See *substeps* on page 10.

<stepxmp>

A step example illustrates a step of a task.

Usage information

The step example can be a couple of words, or an entire paragraph.

Specialization hierarchy

The <stepxmp> element is specialized from the <itemgroup> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<step>
 <cmd>Type a name for the widget.</cmd>
 <stepxmp>For example, <userinput>mywidget</userinput></stepxmp>
</step>
```

<choicetable>

The <choicetable> element contains a series of optional choices available within a step of a task.

Formatting expectations

By default, processors highlight the choice column using bold. To change the highlighting, set the @keycol attribute of the <choicetable> tag to 0 (zero).

Specialization hierarchy

The <choicetable> element is specialized from the <simpletable> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group*, *Display attribute group*, *Simpletable attribute group* (with a narrowed definition of @keycol, given below), *outputclass*, and *spectitle*.

@keycol

On this element, the default value for @keycol is "1". Otherwise, the attribute is the same as defined in *Simpletable attribute group*.

Example

```
<step><cmd>Then this</cmd>
  <substeps>
    <substep importance="optional"><cmd>which is done by doing
 this</cmd></substep>
    <substep importance="required"><cmd>and then this.</cmd></
substep>
  </substeps>
  <choicetable>
    <chhead>
      <choptionhd>Do something</choptionhd>
      <chdeschd>Or Else this</chdeschd>
    </chhead>
    <chrow><choption>Do this</choption>
      <chdesc>and this will happen</chdesc></chrow>
    <chrow><choption>Do that</choption>
      <chdesc>and that will happen</chdesc></chrow>
  </choicetable>
</step>
```

<chhead>

A choice table heading provides heading text for a choice table.

Disposition: / Status:
Original shortdesc follows; we don't need to specify the content model in the shortdesc, and any mention of overriding is moved to the processing section. The <chhead> element is a container inside the <choicetable> element that provides specific heading text to override any default headings for the <choicetable> (such as "Option" and "Description"). The <chhead> element contains both a <choptionhd> and <chdeschd> element as a pair.

Processing expectations

The heading in <chhead> overrides any default headings for the <choicetable> (such as "Option" and "Description").

Specialization hierarchy

The <chhead> element is specialized from the <sthead> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<step><cmd>Then this</cmd>
  <substeps>
    <substep importance="optional"><cmd>which is done by doing
 this</cmd></substep>
    <substep importance="required"><cmd>and then this.</cmd></
substep>
  </substeps>
  <choicetable>
    <chhead>
      <choptionhd>Do something</choptionhd>
      <chdeschd>Or Else this</chdeschd>
```

```
    </chhead>
    <chrow><choption>Do this</choption>
      <chdesc>and this will happen</chdesc></chrow>
    <chrow><choption>Do that</choption>
      <chdesc>and that will happen</chdesc></chrow>
  </choicetable>
</step>
```

<choptionhd>

The <choptionhd> element provides a specific label for the list of options from which a user chooses in order to accomplish a step. The default label for the list of options could be a localized translation of **Option**.

Specialization hierarchy

The <choptionhd> element is specialized from the <stentry> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group*, *outputclass*, and *specentry*.

Example

```
<step><cmd>Then this</cmd>
 <choicetable>
  <chhead>
   <choptionhd>Do something</choptionhd>
   <chdeschd>And this happens</chdeschd>
  </chhead>
  <chrow><choption>Do this</choption>
     <chdesc>and this will happen</chdesc></chrow>
  <chrow><choption>Do that</choption>
     <chdesc>and that will happen</chdesc></chrow>
 </choicetable>
</step>
```

<chdeschd>

A choice description heading provides a specific label for the list of descriptions of options from which a user chooses in order to accomplish a step.

Disposition: / Status:
I think "complete" a step is more appropriate than "accomplish" a step?

Processing expectations

The default label for the list of options could be a localized translation of **Description**.

Specialization hierarchy

The <chdeschd> element is specialized from the <stentry> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group*, *outputclass*, and *specentry*.

Example

```
<step><cmd>Then this</cmd>
  <choicetable>
    <chhead>
      <choptionhd>Do something</choptionhd>
      <chdeschd>Or Else this</chdeschd>
    </chhead>
    <chrow><choption>Do this</choption>
      <chdesc>and this will happen</chdesc></chrow>
    <chrow><choption>Do that</choption>
      <chdesc>and that will happen</chdesc></chrow>
  </choicetable>
</step>
```

<chrow>

The <chrow> element is a container inside the <choicetable> element. The <chrow> element contains both a <choption> and <chdesc> element as a pair.

Specialization hierarchy

The <strow> element is specialized from the <strow> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<step><cmd>Then this</cmd>
  <substeps>
    <substep importance="optional"><cmd>which is done by doing
 this</cmd></substep>
    <substep importance="required"><cmd>and then this.</cmd></
substep>
  </substeps>
  <choicetable>
    <chhead>
      <choptionhd>Do something</choptionhd>
      <chdeschd>Or Else this</chdeschd>
    </chhead>
    <chrow><choption>Do this</choption>
      <chdesc>and this will happen</chdesc></chrow>
    <chrow><choption>Do that</choption>
      <chdesc>and that will happen</chdesc></chrow>
  </choicetable>
</step>
```

<choption>

A choice option describes an option in a choice table that a user could choose to accomplish a step of a task.

Usage information

In a user interface, for example, this might be the name of radio button.

Specialization hierarchy

The <choption> element is specialized from the <stentry> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group*, *outputclass*, and *specentry*.

Example

```
<step><cmd>Then this</cmd>
  <choicetable>
    <chhead>
      <choptionhd>Do something</choptionhd>
      <chdeschd>And this happens</chdeschd>
    </chhead>
    <chrow><choption>Do this</choption>
      <chdesc>and this will happen</chdesc></chrow>
    <chrow><choption>Do that</choption>
      <chdesc>and that will happen</chdesc></chrow>
  </choicetable>
</step>
```

<chdesc>

The <chdesc> element is a description of an option from a choice table that a user chooses while performing a step to accomplish a task. It explains why the user would choose that option and might explain the result of the choice when it is not immediately obvious.

Specialization hierarchy

The <chdesc> element is specialized from the <stentry> element in the topic modules.

Attributes

The following attributes are available on this element: *Universal attribute group*, *outputclass*, and *specentry*.

Example

```
<step><cmd>Then this</cmd>
  <substeps>
    <substep importance="optional"><cmd>which is done by doing
 this</cmd></substep>
    <substep importance="required"><cmd>and then this.</cmd></
substep>
  </substeps>
  <choicetable>
    <chrow><choption>Do this</choption>
```

```
      <chdesc>and this will happen</chdesc></chrow>
    <chrow><choption>Do that</choption>
      <chdesc>and that will happen</chdesc></chrow>
  </choicetable>
</step>
```

<choices>

Choices contain a list of choices for users that need to choose one of several actions while performing the steps of a task.

Specialization hierarchy

The <choices> element is specialized from the <ul> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<step><cmd>Choose a server.</cmd>
<choices>
<choice>If you have a remote server you want to test on, type
 the
IP address or hostname of the server here.</choice>
<choice>If you want to do local testing, just type localhost.</
choice>
</choices>
</step>
```

<choice>

Each <choice> element describes one way that the user could perform the current step.

Specialization hierarchy

The <choice> element is specialized from the <li> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<step><cmd>Choose a server.</cmd>
  <choices>
    <choice>If you have a remote server you want to test on,
 type the
IP address or hostname of the server here.</choice>
    <choice>If you want to do local testing, just type
 localhost.</choice>
  </choices>
```

```
</step>
```

<steptroubleshooting>

Step troubleshooting element provides information that is designed to help remedy the situation when a step does not complete as expected.

Usage information

In particular, this element can be used to explain how users can recover when the results of a step do not match those listed in the <stepresult> element.

Tip: Do not use <note type="trouble"> inside of <steptroubleshooting> because its meaning there would be ambiguous.

Specialization hierarchy

The <steptroubleshooting> element is specialized from the <itemgroup> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

The following example illustrates using the <steptroubleshooting> element with a single action:

```
  <step>
    <cmd>Select <uicontrol>Perform system backup</uicontrol></
cmd>
    <stepresult>
      <p>The message <systemoutput>Backup successfully
        completed</systemoutput> displays.</p>
    </stepresult>
    <steptroubleshooting>
      <p>If an error message displays during the system backup,
        locate the error ID in the <cite>Troubleshooting Guide
        </cite> and follow the instructions there.</p>
    </steptroubleshooting>
  </step>
```

The following example illustrates using the <steptroubleshooting> element with several actions:

```
  <step>
    <cmd>Log in to the system</cmd>
    <stepresult>
      <p>The <wintitle>Welcome</wintitle> screen appears.</p>
    </stepresult>
    <steptroubleshooting>
      <p>If the <wintitle>Welcome</wintitle> screen does not
        appear, try one or more of the following:
        <ul>
          <li>Verify that the user name was entered correctly</
li>
```

```
          <li>Verify that the password was entered correctly</
li>
          <li>Confirm that the maintenance contract is still
            active</li>
        </ul>
      </p>
    </steptroubleshooting>
  </step>
```

<stepresult>

The <stepresult> element provides information on the expected outcome of a step.

Usage information

If a user interface is being documented, the outcome could describe a dialog box opening or the appearance of a progress indicator. Step results are useful to assure a user that they are on track, but should not be used for every step as this quickly becomes tedious.

Specialization hierarchy

The <stepresult> element is specialized from the <itemgroup> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<steps>
 <step>
  <cmd importance="urgent">Once you have the water place it in
 the microwave.</cmd>
  <info>Try not to spill any, as water is very wet.</info>
 </step>
 <step importance="required">
  <cmd>Start the Microwave.</cmd>
  <stepxmp>As an example, push the <b>Start</b> button</stepxmp>
  <stepresult>The Microwave starts running. You should hear it
 humming.</stepresult>
 </step>
 <step importance="optional">
  <cmd>Once the water begins to boil, stop the Microwave.</cmd>
 </step>
</steps>
```

<tutorialinfo>

Tutorial information contains additional information that is useful when the task is part of a tutorial.

Specialization hierarchy

The <tutorialinfo> element is specialized from the <itemgroup> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<steps>
  <step>
    <cmd>Do this</cmd>
    <tutorialinfo>In your editor, open the first element and
 click on
      the dialog.</tutorialinfo>
  </step>
  <step>
    <cmd>Do that</cmd>
    <tutorialinfo>Move the framulator into the foobar box.</
tutorialinfo>
  </step>
</steps>
```

<tasktroubleshooting>

Task troubleshooting information is information that is designed to help users remedy the situation when a task does not complete as expected.

Usage information

In particular, the <tasktroubleshooting> element can be used to explain how users can recover when the results of a task do not match those listed in the <result> element. The troubleshooting remedy typically contains one or more actions for solving a problem. For complex remedies, link to another task.

Formatting expectations

Specialization hierarchy

The <tasktroubleshooting> element is specialized from the <section> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
  <steps>
    <step><cmd>...</cmd></step>
    <!-- ... more steps ... -->
  </steps>
  <result>
    <p>The <uicontrol>User Type</uicontrol> menu updates to
      display the new types you added.</p>
  </result>
  <tasktroubleshooting>
    <p>If the User Type menu does not display the additions, try
      one or more of the following:
      <ul>
        <li>Refresh the page</li>
```

```
        <li>Verify that <wintitle>Add Types</wintitle> is not
          still open; if so, go to it and press
          <uicontrol>OK</uicontrol>.</li>
      </ul>
    </p>
  </tasktroubleshooting>
  <example>
    <p>For example, you could also do xyz.</p>
  </example>
  <postreq>
    <p>Once completed, you need to consider abc.</p>
  </postreq>
```

Tip: Do not use `<note type="trouble">` inside of `<tasktroubleshooting>` because its meaning there would be ambiguous.

<result>

A result describes the expected outcome for the task as a whole.

Usage information

Note: To describe the outcome of a specific step, use the `<stepresult>` element instead.

Specialization hierarchy

The `<result>` element is specialized from the `<section>` element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<task id="sqlj">
  <title>Creating an SQLJ file</title>
  <taskbody>
    <context>Once you have set up SQLJ, you need to create a new
 SQLJ file.
      You cannot add #sqlj statements directly in the Source
 pane of the
      Workbench.</context>
    <result>The SQLJ file is successfully created when the SQLJ
 server
      displays the "File Created" dialog.</result>
  </taskbody>
</task>
```

<postreq>

Post-requisites are steps or tasks that the users should perform after completing the current task.

Usage information

The `<postreq>` element often is supported by hyperlinks to the next task or tasks.

Specialization hierarchy

The <postreq> element is specialized from the <section> element in the topic module.

Attributes

The following attributes are available on this element: *Universal attribute group* and *outputclass*.

Example

```
<postreq>Notify the proctor upon completing this self-test.</
postreq>
```

Index