Review E: Troubleshooting

Table of contents

1 Topic and map document types

The Technical Content package contains various document types: concept, glossary entry, glossary group, reference, general task, strict task, and troubleshooting. The package also includes the bookmap document type.

1.1 Troubleshooting

Troubleshooting topics are designed to document corrective action, such as troubleshooting and alarm clearing.

> **Comment by Stan Doherty**
> Recommend "Troubleshooting topics are designed to document diagnostic and corrective action" . . .
>
> ---
>
> Kris Eberlein, 12 February 2023
>
> Changed to read "Troubleshooting topics are designed to document problems that people might encounter and provide solutions, if possible. They provide a topic structure that enables content authors to describe a problem, provide diagnostic information, discuss causes, and outline possible solutions."
>
> **Disposition: Completed**

Purpose

Troubleshooting topics serve the following purposes:

1. Define the condition or symptom, which usually is an undesirable state in a system, a product, or a service that a reader wants to correct
2. Provide information that helps the reader diagnose which cause and remedy pairs apply

    > **Comment by dstevens73**
    > which cause applies
    >
    > ---
    >
    > Kris Eberlein, 12 February 2023
    >
    > I've recast each list item. See the draft comment after the list.
    >
    > **Disposition: Closed**

3. Explain the cause of the condition or symptom, if it is known, and describe how to remedy the condition or symptom, if possible

> **Comment by Stan Doherty**
> The third list item seems redundant to the first two.
>
> ---
>
> Kris Eberlein, 12 February 2023
>
> I've recast each list item. See the draft comment after the list.
>
> **Disposition: Closed**

Comment by Kristen James Eberlein on 12 February 2023
I've redrafted the ordered list as follows:

1. Describe the problem condition, which usually is a state in a system, product, or service that a reader wants to correct
2. Provide information that helps the reader diagnose the cause of the problem, if it is known
3. Explain the cause of the problem and how to fix it, if possible

Disposition: Completed

Content model

The troubleshooting topic is divided into three parts:

Comment by nancylph

the first and third items in this list mirror the names of the elements we use for them, but the middle one uses 'diagnosis' rather than diagnostics. Do we want to keep this, or not?

Dawn: I think it's fine. the connection seems clear to me.

Disposition: Closed

Condition

This portion of the topic can contain a <condition> (11) element. This element is optional, as often the problematic condition can be adequately described in the title and short description.

Diagnosis

This portion of the topic provides information that helps determine which cause and remedy pair applies to a given situation. It can contain the following elements: <diagnostics-general> (14) and <diagnostics-steps> (17).

Comment by dstevens73
which cause applies

Stan Doherty

. . "which causes and remedies apply" . . .

Kris Eberlein, 12 February 2023

Changed to read "This optional section of the topic provides information about how to determine possible causes of the problem. It is specified by the <tdiagnostics> element, which can contain both general information, including tables and flow charts, as well as steps."

Disposition: Completed

Trouble solution

Comment by Kristen James Eberlein on 07 February 2023

The "definition" here needs to be reworded. It probably should address the following points:

- A troubleshooting topic must contain at least one <troublesolution> element.
- A <troublesolution> must contain either a <cause> or<remedy> element.

- A <remedy> element must include either <steps> or <steps-unordered>.

Dawn: remedy can also include <steps-informal>; in all cases make sure it is <troubleSolution>, with the capital S. Also, I thought we changed the model in 2.0 to not require a <troubleSolution> – it's possible the topic's purpose is just diagnostics.

Kris Eberlein, 12 February 2023

@Dawn, thanks for bringing up those points about the content model of <remedy>, all of which are correct.

Disposition: Unassigned

This portion of the topic provides information about how to remedy the problematic solution. It can contain multiple <cause> (9) and <remedy> (19) pairs.

Comment by nancylph

I don't think we want to remedy a 'problematic solution'. We want to remedy a problem by providing a solution.

Kris Eberlein, 08 February 2023

Too true! Thanks for catching this. Changed "problematic solution" to "problem".

Disposition: Completed

Cause and remedy might occur in combinations other than pairs. It is possible to have:

- Multiple causes with the same remedy
- A single cause with more than one remedy
- A remedy with no known cause
- A cause with no known remedy

Comment by dstevens73
Yes! This is my point throughout about why I was working to eliminate the word "pairs"

Kris Eberlein, 12 February 2023

Marking as "Closed," since no change is required.

Disposition: Closed

Examples

This section of the topics contains examples of troubleshooting topics.

Figure 1: Simple troubleshooting topic

The following code sample shows a simple troubleshooting topic. The title and short description describe the problem, and the single <troublesolution> element contains explanations of the cause of the problem and how to remedy it.

Comment by Kristen James Eberlein on 12 February 2023
<troubleSolution>, not <troublesolution>

Disposition: Completed

```
<troubleshooting id="oasis-spec-not-rendered-correctly">
  <title>Specification PDF is not rendered correctly</title>
  <shortdesc>The specification URIs and notices appear in the TOC; they also appear
             twice in the body of the document.</shortdesc>
  <troublebody>
    <troubleSolution>
      <cause>
        <p>This problem occurs when the <xmlelement>notices</xmlelement> element
           for external publishing is not excluded.</p>
      </cause>
      <remedy>
        <steps>
          <step>
            <cmd>Use a DITAVAL file that excludes the <xmlelement>notices
                 platform="external-publishing-engine"</xmlelement> element
                 when you generate the PDF.</cmd>
          </step>
        </steps>
      </remedy>
    </troubleSolution>
  </troublebody>
</troubleshooting>
```

Figure 2: Complex troubleshooting topic

The following code sample shows a complex troubleshooting topic about "Blinking printer lights." It contains a `<diganostics-general>` elements that contains a table that outlines printer light conditions and the associated remedies. It also includes several `<troubesolution>` elements that reuses steps from other DITA topics.

Comment by nancylph

Couple of things: 1) the diagnostics-general element is mis-spelled. 2) There is only a single instance of that element, so it should say 'element' not 'elements' 3) the troublesolution element is also mis-spelled and 4) also has a disagreement between the multiple instances and the singular 'reuses' verb.

Kris Eberlein, 12 February 2023

@Nancy, thanks for noting these problems. I've corrected all the errors.

Disposition: Completed

Comment by Stan Doherty

The phrase "associated remedies" assumes that we writers are fairly confident that Cause-X is closely associated with Remedy-Y. It may be prudent to go with "possible remedies".

Kris Eberlein, 12 February 2023

Done.

Disposition: Completed

Comment by dstevens73

In the example, the remedies "clean" and "ink" do not have explicit conrefs to show the reuse, they just appear to be empty steps.

Kris Eberlein, 12 February 2023

I've fixed the problems.

Disposition: Completed

```
troubleshooting id="blinking-lights">
  <title>Blinking printer lights</title>
  <troublebody>
  <condition><p>The indicator lights are blinking and you are unable to print.</p></condition>
  <diagnostics>
    <diagnostics-general><p>Use the following table to diagnose the problem.</p>
      <simpletable frame="all" id="light-diagnostics" relcolwidth="1* 2*">
        <sthead>
          <stentry>Lights</stentry>
          <stentry>Issue</stentry>
        </sthead>
        <strow>
          <stentry>The power light is flashing and the resume light is off./stentry>
          <stentry>The printer is preparing to print. No action is required. The light
                   will stop flashing when the printer has received all data.</stentry>
        </strow>
        <strow>
          <stentry> The connecton and error lghts flash for five seconds.</stentry>
          <stentry>The printer has lost connection with the camera. Unplug and replug the
                   camera.</stentry>
        </strow>
        <strow>
          <stentry> The power light is on, and the resume light is flashing.</stentry>
          <stentry>Printer jam. See <xref href="#./jam"/>.</stentry>
        </strow>
        <strow>
          <stentry>The power and resume lights are flashing.</stentry>
          <stentry>Printer is stalled and needs to be restarted. See
                   <xref href="#./restart"/>.</stentry>
        </strow>
        <strow>
          <stentry>The left cartridge light is on, and right cartridge list is off.</stentry>
          <stentry>Low ink. See <xref href="#./ink"/>.</stentry>
        </strow>
        <strow>
          <stentry>The left cartridge light is on, and the right cartridge list is flashing.
          </stentry>
          <stentry>Dirty ink cartridge. See <xref href="#./clean"/>.</stentry>
        </strow>
        <strow>
          <stentry>The connection list is on, and the error light is flashing.</stentry>
          <stentry>The camera is not set to the correct mode for transferring photos. Change
                   the camera mode.</stentry>
        </strow>
      </simpletable>
    </diagnostics-general>
  </diagnostics>
  <troubleSolution>
    <remedy id="jam">
      <title>Clearing a paper jam </title>
      <steps conkeyref="clear-jam/steps">
        <step><cmd/></step>
      </steps>
    </remedy>
  </troubleSolution>
  <troubleSolution>
    <remedy id="clean">
      <title>Cleaning ink cartridge</title>
      <steps>
        <step><cmd/></step>
      </steps>
    </remedy>
  </troubleSolution>
  <troubleSolution>
    <remedy id="ink">
      <title>Replacing ink cartridge</title>
      <steps>
        <step><cmd/></step>
      </steps>
    </remedy>
    </troubleSolution>
  </troublebody>
</troubleshooting>
```

Comment by Kristen James Eberlein on 12 February 2023
Maybe we should include a screen capture of the table. We've been doing that consistently for tables, and they've also proved useful in helping us capture problems with the content of the code sample.

Kris Eberlein, 13 February 2023

Done.

Disposition: Completed

Related reference

Troubleshooting elements (9)
Troubleshooting elements provide the fundamental structure for troubleshooting topics. Troubleshooting topics outline problem conditions and aid readers in diagnosing and remedying the situation.

Comment by Kristen James Eberlein on 10 February 2023

Suggest changing the 2nd sentence to "Troubleshooting topics describe problems and provide information about how to fix them."

Disposition: Completed

2 Troubleshooting elements

Troubleshooting elements provide the fundamental structure for troubleshooting topics. Troubleshooting topics outline problem conditions and aid readers in diagnosing and remedying the situation.

Comment by Kristen James Eberlein on 10 February 2023

Suggest changing the 2nd sentence to "Troubleshooting topics describe problems and provide information about how to fix them."

Disposition: Completed

Related concepts

Troubleshooting (3)
Troubleshooting topics are designed to document corrective action, such as troubleshooting and alarm clearing.

Comment by Stan Doherty
Recommend "Troubleshooting topics are designed to document diagnostic and corrective action" . . .

Kris Eberlein, 12 February 2023

Changed to read "Troubleshooting topics are designed to document problems that people might encounter and provide solutions, if possible. They provide a topic structure that enables content authors to describe a problem, provide diagnostic information, discuss causes, and outline possible solutions."

Disposition: Completed

2.1 <cause>

The `<cause>` element describes a potential source of the problem that is described in the troubleshooting topic.

Comment by Kristen James Eberlein on 12 February 2023
Changed to "The `<cause>` element describes a potential source of the problem.
Disposition: Completed

Usage information

The `<cause>` element is a component of a cause and remedy pair. The cause might be omitted if it is implicit or if the remedy is not associated with a cause.

Comment by dstevens73
a component of <troubleSolution>

Kris Eberlein, 12 February 2023

Changed to read "The `<cause>` element is a component of a potential solution."

Disposition: Completed

This information should be brief. If it is not possible to make the information brief, this might be a sign that a full troubleshooting topic is needed.

Comment by nancylph

I find this a bit confusing. My impression is that a cause can only appear in a troubleshooting topic (I had thought it could appear elsewhere, but that doesn't seem to be the case from the 1.3 spec) If it can only appear in a troubleshooting topic, then why would a 'full troubleshooting topic' only be required for it sometimes?

Kris Eberlein, 09 February 2023

I agree. I think the paragraph should be deleted. I think that such information was originally written for the embedded troubleshooting elements: `<note type="trouble>`, `<steptroubleshooting>`, and `<tasktroubleshooting>`. It does make sense for those elements.

I've deleted the paragraph from the DITA source.

Disposition: Completed

Specialization hierarchy

The `<cause>` element is specialized from `<section>`. It is defined in the troubleshooting module.

Attributes

The following attributes are available on this element: universal attributes (41).

Example

In the following code sample, the `<cause>` element contains information that explains the origins of the problem:

Comment by nancylph

under the <cause> section below, the word 'the' is repeated, once at the end of the first line, and then at the beginning of the second

---

Kris Eberlein, 09 February 2023

Fixed.

Dawn: This wouldn't have needed fixing if this was either just a reference to the main troubleshooting topic or a conref of the section it is repeating.

Kris Eberlein, 12 February 2023

Yes, we could simply have a cross reference to the `<troubleshooting>` topic. That strategy has two potential disadvantages:

- It forces the reader to read through the XML source for an entire topic, rather than seeing a smaller snippet: where the `<cause>` element fits within `<troubleSolution>`.
- It does not allow us to use the introductory paragraph to provide more context about the `<cause>` element.

Yes, we could conref a <codeblock> element. However, then we cannot highlight the element that is being showcased.

Disposition: Completed

Comment by Stan Doherty
Note that the content in the <cause> example contains a suggested solution. Premature. If, based on what is in <cause>, the customer reseats the board and something gets fried – my company is on the hook to replace/service the board for free. Support teams want customers to have maximum context before performing a corrective action. Separating cause analysis from remedies is a best practice. This example appears in multiple sections.

Kris Eberlein, 12 February 2023

@Stan, I think you are raising several points here, so I'll list them in an ordered list (in order to make talking about the points easier):

1. **Quality of examples.** Here in the spec, we need to straddle the line between "realistic examples" and examples that are simple enough for the spec reader to grasp. Obviously, this is especially difficult for troubleshooting topics, which can be complex both in structure and in content.

   This example appeared in the DITA 1.3 spec, however, it did use the <responsibleParty> element to indicate that reseating the board is a task that should be performed by a maintenance technician. Would you be more comfortable if this example added that element back? Or do you want to create an alternate example?
2. **Separating causes from remedies.** Is your concern with the 2nd sentence in the <cause> element? We could remove it. I do want to comment that the very nature of the <troubleSolution> element suggests that there is a relationship between a cause and a remedy.
3. **Modeling best practices for troubleshooting topic content.** While we don't want to have any egregious content in spec examples, we really cannot expect these topics to model authoring best practices. That is always going to be something that an individual company needs to decide for their content authors.

Disposition: Open

```
<troubleSolution>
  <cause>A recurring memory fault indicates a possible problem with the
         the system memory board. Reseating the board might fix the problem.
  </cause>
  <remedy>
    <steps conref="reseatBoard.dita#reseatBoard/steps">
      <step><cmd/></step>
    </steps>
  </remedy>
</troubleSolution>
```

2.2 <condition>

The <condition> element describes a state that the troubleshooting topic is intended to remedy. This information helps the user decide whether a troubleshooting topic might contain an applicable remedy for a problem.

Usage information

This section should add to or clarify information that is in the title or short description of the troubleshooting topic. If the title and short description adequately describes the condition, this element might be omitted.

Specialization hierarchy

The `<condition>` element is specialized from `<section>`. It is defined in the troubleshooting module.

Attributes

The following attributes are available on this element: universal attributes (41).

Example

In the following code sample, the `<condition>` element contains information that elaborates on the information provided by the title and short description:

Comment by Stan Doherty
Nit - hardware documentation differentiates between a system receiving power and it powering up. Just flicking the power switch will "turn it on", but it may not power up across all its internal components.

Kris Eberlein, 12 February 2023

Can you suggest wording that would handle your nit?

FYI, this was an example topic in the "DITA 1.3 Troubleshooting" whitepaper.

Kris Eberlein, 14 February 2023

Based on e-mail exchange with Stan, changed "the power switch is on" to "the system is powered up".

Disposition: Completed

```
<troubleshooting id="system-will-not-turn-on">
  <title>System will not turn on</title>
  <shortdesc>Everything looks right, but the system still does not start.</shortdesc>
  <troublebody>
    <condition>
      <title>Condition</title>
      <p>The system is plugged in, the power switch is on, but the system does not start.</p>
    </condition>
    <troubleSolution>
      <!-- ... -->
    </troubleSolution>
  </troublebody>
</troubleshooting>
```

Alternately, the short description could be enhanced and the `<condition>` element eliminated:

```
<troubleshooting id="system-will-not-turn-on">
  <title>System will not turn on</title>
  <shortdesc>he system is plugged in, the power switch is on, but the system does not start.</
shortdesc>
```

```
 Comment by
              dstevens73
Missing "T" at the beginning of shortdesc
```

Kris Eberlein, 13 February 2023

Fixed.

Disposition: Completed

```
  <troublebody>
    <troubleSolution>
      <!-- ... -->
    </troubleSolution>
  </troublebody>
</troubleshooting>
```

Comment by Stan Doherty
The next paragraph seems introductory to troubleshooting in general. Not particularly detailed or relevant to <condition>.

Kris Eberlein, 11 February 2023

I agree that the paragraph is applicable to the troubleshooting topic in general. However, I think it is germane here because the example presents two different ways of describing the condition:

- Title and short description
- Title, short description, and `<condition>`

In my experience, an implementation usually picks one of these two approaches.

Disposition: Closed

How implementations choose what markup patterns they use might depend on their stylesheets and how they deliver troubleshooting information to readers.

2.3 <diagnostics>

The `<diagnostics>` element contains information that helps readers determine the cause of a problem and what remediation steps are applicable.

Comment by nancylph

'the cause of the problem' is concise and pithy; 'what remediation steps are applicable' is much less so. So putting them together strikes my eye/ear as just a bit strange. It couldn't simply be 'the cause of a problem and how to fix it'?

Kris Eberlein, 09 February 2023

It would be nice to simply say "the cause of a problem and how to fix it," but sometimes there won't actually be a remedy :(

Dawn: technically, I think we could leave out everything after the cause of the problem. Or maybe fix nancy's observation by saying "the cause of a problem, and if available, how to fix it"

Kris Eberlein, 10 February 2023

I've changed this to "The `<diagnostics>` element contains information that helps readers determine the cause of a problem."

Disposition: Completed

Comment by Stan Doherty

Not sure how to nuance it here, but one of the main justifications for teams developing troubleshooting docs is preventing calls to Support. Relative to that goal, any diagnostic section of a TS topic is there to assist the customer is interpreting what she or he is seeing – messages, flashing LEDs, terrible performance, etc.. It's actually a teaching moment. Having this element as part of the DITA troubleshooting domain is really important. Investing in formal root cause analysis is interpretive and, at $400 a Support call, can save lots of money.

Kris Eberlein, 13 February 2023

I agree with you 100% here, @Stan. I think this information should be expanded as a topic for the troubleshooting committee note.

Disposition: Closed

Usage information

Diagnostic information is useful when more than one possible cause and remedy pairs are applicable. The `<diagnostics-general>` element permits text-based content, including tables and flowcharts, while the `<diagnostics-steps>` element allows for the use of the `<steps>` element.

Comment by dstevens73
when more than one possible cause is applicable? again, the "pairs" is not entirely true, and the diagnostics, I believe, is to determine which cause applies. the remedy is then nice to also have, but its not the direct thing the diagnosis is determining.

Kris Eberlein, 10 February 2023

Changed the first sentence to read: "Diagnostic information is useful when there is more than one potential cause associated with a symptom."

Disposition: Completed

Specialization hierarchy

The `<diagnostics>` element is specialized from `<bodydiv>`. It is defined in the troubleshooting module.

Attributes

The following attributes are available on this element: universal attributes (41).

Example

See `<diagnostics-general>` (14) and `<diagnostics-steps>` (17).

2.4 <diagnostics-general>

The `<diagnostics-general>` element includes non-procedural information that can help determine which of multiple cause and remedy pairs apply to a specific situation. Results of the diagnostics might link to the specific cause and remedy pair that applies to each result.

Comment by dstevens73
which of multiple causes apply. [apologies if my first comment is determined to be irrelevant, but I thought I should be marking every place "pair" is used.]

Kris Eberlein, 10 February 2023

Changed to "The `<diagnostics-general>` element includes non-procedural information that can help determine the cause of the symptom. Results of the diagnoses might link to a possible solution."

Disposition: Completed

Usage information

This element is useful for presenting non-procedural diagnostic information, for example, a diagnostic table or a flowchart. Non-procedural diagnostic information can be used when the symptoms can be observed and do not require people to take action.

Specialization hierarchy

The `<diagnostics-general>` element is specialized from `<section>`. It is defined in the troubleshooting module.

Attributes

The following attributes are available on this element: universal attributes (41).

Example

The following code sample shows how the `<diagnostics-general>` element can contain a table to help a user determine which cause and remedy pair is applicable.

Comment by dstevens73
which cause is applicable. The table then points the user to the corresponding remedy.

Kris Eberlein, 12 February 2023

Changed the introductory paragraph to read as follows: "The following code sample shows how the `<diagnostics-general>` element can contain a table to help a reader determine the cause of a problem. The table then references the associated remedy."

Disposition: Completed

```
<troubleshooting id="car-makes-funny-noises">
  <title>Car is making funny noises.</title>
  <shortdesc>You probably know how your vehicle sounds when it’s running
    properly. Listening to your car can help you troubleshoot problems. If
    you hear a strange sound, pay attention and react
    accordingly.</shortdesc>
  <troublebody>
    <diagnostics>
      <diagnostics-general>
        <simpletable frame="all" relcolwidth="1* 1*">
          <sthead>
            <stentry>Symptom</stentry>
            <stentry>Probable cause</stentry>
          </sthead>
          <strow>
            <stentry>Clunking noise on bumps only</stentry>
            <stentry><xref href="#./checkstruts"/></stentry>
          </strow>
          <strow>
            <stentry>Continuous clunking noise</stentry>
            <stentry><xref href="#./checkballjoints"/></stentry>
          </strow>
          <strow>
            <stentry>Ticks when in neutral</stentry>
            <stentry><xref href="#./checkexhaust"/></stentry>
          </strow>
          <strow>
            <stentry>Ticks only in reverse</stentry>
            <stentry><xref href="#./checkbrakes"/></stentry>
          </strow>
          <strow>
            <stentry>Ticks in turns and curves</stentry>
            <stentry><xref href="#./checkcvjoint"/></stentry>
          </strow>
          <strow>
            <stentry>Ticks only when cold</stentry>
            <stentry><xref href="#./checkcatalyticconverter"/></stentry>
          </strow>
          <strow>
            <stentry>Ticks only at slow speed</stentry>
            <stentry><xref href="#./checkwheels"/></stentry>
          </strow>
        </simpletable>
      </diagnostics-general>
    </diagnostics>
    <!-- The rest of this topic contains <troublesolution> elements, each of which
         contains a remedy. The cross references in the above steps resolve to the
         <remedy> elements. -->
  </troublebody>
</troubleshooting>
```

The table in the `<diagnostic-general>` element might be rendered in the following way. The hyperlinks in the "Probable cause" column resolve to the `<remedy>` elements in the topic.

Comment by nancylph

in the above paragraph, the element is diagnostics-general, not diagnostic-general

---

Kris Eberlein, 09 February 2023

Fixed.

Disposition: Completed

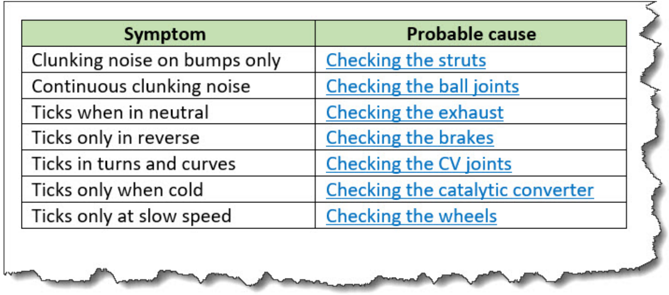
Comment by Stan Doherty
Not a realistic example outside dialing into "Car Talk". :-) Typically there is a very specific symptom and a range of possible root causes for that symptom. Hearing strange noises in my old car, at least, is not deterministic, persistent, or worthy of a troubleshooting topic. I'll bet that we could do a bang-up example on why a particular conkeyref does not resolve.

Kris Eberlein, 10 February 2023

Sigh. What content we use for examples is tricky. Is this realistic for a book about taking care of your old car? Probably not. Is the content something that someone reading this spec topic can understand? Probably yes.

As we've moved to trying to have "realistic examples," I think our first directive was to get rid of the "Do this … Then do that …" type of stuff. In comparison to those sort of examples, this one is very "realistic", and it probably took Dawn a few hours to put it together. And I know I spent at least an hour finessing the line breaks and creating the screen capture.

Disposition: Referred



Symptom	Probable cause
Clunking noise on bumps only	Checking the struts
Continuous clunking noise	Checking the ball joints
Ticks when in neutral	Checking the exhaust
Ticks only in reverse	Checking the brakes
Ticks in turns and curves	Checking the CV joints
Ticks only when cold	Checking the catalytic converter
Ticks only at slow speed	Checking the wheels

Comment by Kristen James Eberlein on 09 February 2023

I think we need to change the screen capture and the sample mark up that it is based on. For me, there is a disconnect between the column header "Probable cause" and the remedy titles: "Checking the struts," etc.

Perhaps the contents of the 2nd column should read more like "Struts. Go to "Checking the struts"?

Dawn: agreed. I like your proposed solution.

Kris Eberlein, 13 February 2023

Done.

Disposition: Completed

2.5 <diagnostics-steps>

The `<diagnostics-steps>` element includes step-by-step information that can help readers determine which cause and remedy pairs apply. Results of each diagnostic step might link to the specific cause and remedy pair that is applicable.

Comment by dstevens73
which cause applies. Results of each diagnostic step might link to the specific remedy that is applicable.

Kris Eberlein, 12 February 2023

Changed to read "The `<diagnostics-steps>` element includes step-by-step information that can help readers determine the cause of the symptom . Results of the diagnostic steps might link to potential solutions."

Disposition: Completed

Usage information

This element is helpful for situations where the reader must perform a series of steps to determine the cause of the problem.

Specialization hierarchy

The `<diagnostics-steps>` element is specialized from `<section>`. It is defined in the troubleshooting module.

Attributes

The following attributes are available on this element: universal attributes (41).

Example

The following code sample shows how the `<diagnostic-steps>` element can provide step-by-step instructions to help someone determine which cause and repair pair is applicable to their problem:

Comment by dstevens73
"which cause applies to their problem and therefore which remedy will solve the problem." In this case, the sample topic implies in the comment section that the <troubleSolution> contains just the remedy element, not a pair. [and yes, I'm pretty sure I wrote these in the first place...]

Kris Eberlein, 12 February 2023

@Dawn, many thanks for your rewording suggestions! I've changed this to read "The following code sample shows how the `<diagnostic-steps>` element can provide step-by-step instructions to help someone determine the cause of a problem and the potential solution:"

Disposition: Completed

Comment by nancylph

the element in the paragraph above should be diagnostics-steps, not diagnostic-steps

---

Kris Eberlein, 09 February 2023

Fixed.

Disposition: Completed

```
<diagnostics>
  <diagnostics-steps>
    <steps>
      <step>
        <cmd>Open the command prompt and type <userinput>ipconfig</userinput>.</cmd>
        <info>The Default Gateway (listed last) is your router’s IP. Your computer’s IP
              address is the number next to “IP Address.” If your computer’s IP address
              starts with 169, the computer is not receiving a valid IP address. See
              <xref href="#./ipaddress"/>.
        </info>
      </step>
      <step>
        <cmd>If your address does not start with 169, type
             <userinput>tracert8.8.8.8</userinput> to view each step between your router
             and the Google DNS servers.</cmd>
        <info>If the error comes up early along the pathway, see
              <xref href="#./resetnetwork"/></info>
      </step>
      <step>
        <cmd>If everything is working with Google, use the <cmdname>nslookup</cmdname>
             command to determine if there's a problem with the server you are trying
             to connect to.</cmd>
        <info>If you received results such as <msgph>Timed Out</msgph>,
              <msgph>Server Failure</msgph>, <msgph>Refused</msgph>,
              <msgph>No Response from Server</msgph>, or
              <msgph>Network is unreachable<msgph>, the problem originates in the DNS
              server for your destination.</info>
      </step>
      <step>
        <cmd>If the previous steps turn up no problems, contact your ISP to see if they're
               having issues.</cmd>
      </step>
    </steps>
  </diagnostics-steps>
</diagnostics>
  <!-- The rest of this topic contains two <troublesolution> elements, each of which contains
       a remedy. One remedy provides instructions for "Resetting your IP address" and the
       other provides instructions for "Resetting your local network". The cross references
       in the above steps resolve to the <remedy> elements. -->
```

2.6 <remedy>

The `<remedy>` element contains steps that are a potential solution for the problem described in the troubleshooting topic.

Comment by Kristen James Eberlein on 12 February 2023
Changed to read "The `<remedy>` element contains steps that are a potential solution for the problem."
Disposition: Completed

Comment by Kristen James Eberlein on 12 February 2023
The short description should maybe mention that <remedy> can contain <responsibleParty>. I added the following sentence: "In addition, it also contains information about the person or team who should perform the task."
Disposition: Completed

Usage information

The `<remedy>` element is a component of a cause and remedy pair. The remedy might be omitted if there is no known remedy for the cause.

Comment by dstevens73
component of a <troubleSolution> element.

Kris Eberlein, 12 February 2023

Changed the first sentence to read "The `<remedy>` element is a component of potential solution."

Disposition: Completed

Specialization hierarchy

The `<remedy>` element is specialized from `<section>`. It is defined in the troubleshooting module.

Attributes

The following attributes are available on this element: universal attributes (41).

Example

In the following code sample, the `<remedy>` element contains instructions for how the responsible party can fix the problem. Note that rather than cross-referencing to another topic, the `<steps>` element from an existing topic is reused.

Comment by nancylph

under the <cause> section below, the word 'the' is repeated, once at the end of the first line, and then at the beginning of the second

---

Dawn: should this just be a reference to the main example? or a conref of the section from that example?

Kris Eberlein, 14 February 2023

Replaced the example with another one that more clearly shows the remedy in the context of the troubleshooting topic.

Disposition: Completed

```
<troubleSolution>
  <cause>A recurring memory fault indicates a possible problem with the
         the system memory board. Reseating the board might fix the problem.
  </cause>
  <remedy>
    <steps conref="reseatBoard.dita#reseatBoard/steps">
      <step><cmd/></step>
    </steps>
  </remedy>
</troubleSolution>
```

2.7 < responsibleParty>

The `<responsibleParty>` element identifies the individual or team whose task it is to perform a remedy procedure.

Specialization hierarchy

The `<responsibleParty>` element is specialized from `<p>`. It is defined in the troubleshooting module.

Comment by Kristen James Eberlein on 07 February 2023

Why is this specialized from `<p>`? I know that we included the element because it was part of the IBM troubleshooting specialization, but I think it should be specialized from `<data>`, so it is not rendered by default. It's metadata that implementations might use to generate some text, but I can't imagine that anyone wants it rendered by default.

Dawn: All my clients who use it do want it rendered, if that counts for anything. They want readers of the content to understand that there are some things they should not attempt on their own – they don't have the right permissions, it voids their warranty, etc. We tend to implement special formatting on it – making it look like a note or similar to call attention to the restriction at the beginning of the remedy section.

@Dawn, I'm sure that your clients want something rendered from the `<responsibleParty>` element. But, do they ever render just the contents of the element? Or are they using the element to contain more than the name of the responsible party?

Given the example in this topic, if it was rendered as-is to HTML, there would **just** be the following preceding the steps:

```
<p>IT technical support</p>
```

Can you provide some examples of what your clients put in this element?

Kris Eberlein, 12 February 2023

I wonder if we should redesign this element, if it is intended to be rendered. New name, new definition, new examples. In particular, maybe we should broaden the element to not just include the name of the "responsible party," but also any needed warnings or notes ...

Disposition: Referred

Attributes

The following attributes are available on this element: universal attributes (41).

Example

In the following code sample, the `<responsibleParty>` element indicates the person who should perform the task:

```
<remedy>
  <title>Remedy</title>
  <responsibleParty>IT technical support</responsibleParty>
  <steps>
    <step><cmd>Turn the system power switch to <hwcontrol>Off</hwcontrol>.</cmd></step>
    <step><cmd>Reset the circuit breaker.</cmd></step>
    <step><cmd>Turn the system power switch to <hwcontrol>On</hwcontrol>.</cmd></step>
  </steps>
</remedy>
```

2.8 <troublebody>

The `<troublebody>` element contains the main content of the troubleshooting topic. The troubleshooting body can contain information about the condition, diagnostic information, and one or more possible solutions.

Comment by Kristen James Eberlein on 12 February 2023

Changed the 2nd sentence to read "The troubleshooting body can contain information about the condition, how to diagnose the cause, and one or more possible solutions."

Disposition: Completed

Specialization hierarchy

The `<troublebody>` element is specialized from `<body>`. It is defined in the troubleshooting module.

Attributes

The following attributes are available on this element: universal attributes (41).

Example

See <troubleshooting> (23).

2.9 <troubleshooting>

The <troubleshooting> element is the top-level element for a troubleshooting topic. Troubleshooting topics provide information that enables readers to identify, diagnosis, and potentially remedy a problem situation.

Comment by nancylph

diagnose, not diagnosis

Kris Eberlein, 08 February 2023

Thanks for catching this. I've corrected it. And changed the 2nd sentence to read "Troubleshooting topics provide information that enables readers to identify a condition, diagnose a cause, and potentially fix the problem."

Disposition: Completed

Usage information

Troubleshooting topics begin with a description of a condition that the reader might want to correct, followed by one or more cause and remedy pairs. Each cause and remedy pair is a potential solution to the problem that is described in the condition. Troubleshooting topics represent the kind of information that users typically consult to fix a problem.

Comment by dstevens73

Is everyone ok with "pairs" – technically, it doesn't have to be a pair. <troubleSolution> can contain one or more of <cause> and <remedy>. maybe "combinations" would be more accurate? that said, I like pairs, it's just not technically accurate.

Kris Eberlein, 12 February 2023

I went and looked at the DITA 1.3 topic. The current content of "Usage information" is the 1.3 content plus a sentence that was in an untitled section. I think it needs to be reworked to cover the DITA 2.0 changes (addition of diagnostics, making the <troubleSolution> element optional) and the more precise wording that reviewers think is needed.

I've (tentatively) changed the content of the section to read "Troubleshooting topics begin with a description of a problem that the reader might want to correct. This can be followed by diagnostic information and possible solutions to the problem."

Disposition: Completed

Comment by Stan Doherty

FWIW – diagnosis precedes root cause analysis (RCA). Why are we recommending that RCA precede diagnosis? The truth is that some of the more simple troubleshooting issues can begin with an obvious RCA. Most do not.

Kris Eberlein, 10 February 2023

Is there a particular place in this topic (or collection of topics) where we recommend that root cause analysis (RCA) precedes diagnosis? An example would really help me respond to your comment.

Marking this as "Referred"; I need more info from Stan before I can resolve this comment.

Kris Eberlein, 14 February 2023

Marking this as "Closed," as we do not need to make any changes to the source.

Disposition: Closed

Specialization hierarchy

The `<troubleshooting>` element is specialized from `<topic>`. It is defined in the troubleshooting module.

Attributes

The following attributes are available on this element: architectural attributes (46) and universal attributes (41).

For this element, the `@id` attribute is required.

Example

The following code sample contains a troubleshooting topic. The troubleshooting topic contains three cause and remedy pairs that direct the user to perform sequential troubleshooting tasks: Resetting the alarm, reseating the system memory board, and replacing the memory board. Note that some steps are reused from other topics.

Comment by dstevens73
in this case, the word pair is accurate. there is a one to one relationship in each troubleSolution

---

Kris Eberlein, 12 February 2023

I've changed the second sentence to read: "The troubleshooting topic contains three `<troubleSolution>`elements that direct the user to perform sequential troubleshooting tasks: …"

Disposition: Completed

```
<troubleshooting id="E247" xml:lang="en-us">
  <title><msgph><msgnum>E247</msgnum>: Memory fault has occurred</msgph></title>
  <shortdesc>The system has detected a memory problem.</shortdesc>
  <troublebody>
    <condition>
      <p>The fault indicator flashes on the front panel, and the error log
        contains the <msgnum>E247</msgnum> error message.</p>
    </condition>
    <troubleSolution>
      <cause>p>A transient memory fault has occurred.</p></cause>
      <remedy>
        <steps>
          <step><cmd>Reset the alarm.</cmd></step>
          <step><cmd>Monitor the system periodically to see whether the alarm
              recurs.</cmd>
          </step>
        </steps>
      </remedy>
    </troubleSolution>
    <troubleSolution>
```

```
      <cause><p>A recurring memory fault indicates a possible problem with the
          system memory board. Reseating the board might fix the
          problem.</p></cause>
      <remedy>
        <steps conref="boardReseat.dita#boardReseat/steps">
          <step><cmd/></step>
        </steps>
      </remedy>
    </troubleSolution>
    <troubleSolution>
      <cause>p>Reseating the system memory board did not fix the problem.
          Replacing the board might fix the problem.</p></cause>
      <remedy>
        <steps conref="boardReplace.dita#boardReplace.dita/steps">
          <step><cmd/></step>
        </steps>
      </remedy>
    </troubleSolution>
  </troublebody>
</troubleshooting>
```

Comment by dstevens73
I know that someone asked the question last time about the logic of when elements are shown in context and when they are not. My instinct on this example was to suggest adding diagnostics here and making on the solutions have a responsible party, so all elements refer to the one master example. Since most of the elements (except troubleBody) just show a portion of this example, is it necessary here? Should the extracts in the other topics be conrefs?

Kris Eberlein, 12 February 2023

I don't think we can have one troubleshooting example that we use for all elements. Reasons:

- We'd have to develop an entirely new topic that included both <diagnostics-general> and <diagnostics-steps>.
- The topic would be extremely long, and it would difficult for readers to locate the specific markup. It might perhaps be too long for the PDF output; if the contents of a code block covers more than a full page, our style sheets don't handle it well.

Kris Eberlein, 14 February 2022

I have updated this example to include <responsibleParty> elements.

Disposition: Completed

2.10 <troubleSolution>

The <troubleSolution> element contains a possible cause and remedy pair for the problem that is described in a troubleshooting topic.

Comment by dstevens73
here, more than previously, I do think pair is misleading.

Kris Eberlein, 12 February 2022

Changed to read: "Each <troubleSolution> element contains information about the cause of a problem and a potential remedy."

Disposition: Completed

Usage information

Both the cause and remedy are optional. The cause might be omitted if it is implicit or if the remedy is not associated with a cause. The remedy might be omitted if there is no known remedy for the cause.

Comment by Kristen James Eberlein on 12 February 2023

Changed the first sentence to read as follows: "The troubleshooting topic can contain multiple `<troubleSolution>` elements. A `<troubleSolution>` element can contain a cause, a remedy, or a cause and remedy pair.

Disposition: Completed

Specialization hierarchy

The `<troubleSolution>` element is specialized from `<bodydiv>`. It is defined in the troubleshooting module.

Attributes

The following attributes are available on this element: universal attributes (41).

Example

In the following code sample, the `<troubleSolution>` element contains a cause and remedy pair:

Comment by dstevens73
here "pair" is accurate.

Kris Eberlein, 12 February 2022

No changes required, so marking this comment as "Closed".

Disposition: Closed

Comment by Stan Doherty
System alarms tend to be hardware-oriented and serious – bad mojo with LED lights and mini-console reports. Error messages tend to be generated from software apps and appear in logs or console alerts. Not sure what "Reset the alarm" means here.

Kris Eberlein, 12 February 2022

@Stan, this is from the example that appeared in the DITA 1.3 spec. Are you suggesting that we replace this example topic with another one? If so, are you willing to create the example topic? It should contain at least a condition and three <troublesolution> elements.

Kris Eberlein, 14 February 2023

From an e-mail exchange with Stan:

> This example just happens to be bent - if we're being rigorous. If it is exemplifying how to troubleshoot a physical hardware procedure, it should focus on LEDs, reset switches, and hardware mini-console messages. If it is a software troubleshooting procedure, it should talk about log entries, error messages, console messages, and/or built-in diagnostic commands. I am not religious about the examples at this point. The current example models useful XML markup if not useful technical content. OK to leave it.

Marking this comment as "Closed".

Disposition: Closed

```
<troubleSolution>
  <cause>The <msgnum>E247</msgnum> error message is generated due to a
         transient memory fault.</cause>
  <remedy>
    <steps>
      <step><cmd>Reset the alarm.</cmd></step>
      <step><cmd>Monitor the system periodically to see whether the alarm
              recurs.</cmd>
      </step>
    </steps>
  </remedy>
</troubleSolution>
```

Comment by dstevens73
why wouldn't this just refer to the larger example in troubleshooting? should this at least be a conref

---

Kris Eberlein, 12 February 2023

Marking this as a "Duplicate" comment.

Disposition: Closed

A Aggregated RFC-2119 statements

This appendix contains all the normative statements from the DITA for Technical Content 2.0 specification. They are aggregated here for convenience in this non-normative appendix.

B Attributes

This section contains definitions for commonly-used attributes. If an attribute is defined differently on a specific element, that information is covered in the topic for the specific element.

> **Comment by Kristen J Eberlein on 29 December 2021**
>
> Add a brief overview of the fact that some specific attributes are overloaded – and have different meanings depending on what element they are specified upon.
>
> **Disposition: Unassigned**

B.1 Attribute groups

Many of the attributes used on DITA elements are defined in attribute groups. These attribute groups are used both in the grammar files and the specification,

Architectural attributes

This group contains a set of attributes that are defined for document-level elements such as `<topic>` and `<map>`.

@DITAArchVersion (architectural attributes)

Specifies the version of the DITA architecture that is in use. This attribute is in the namespace `http://dita.oasis-open.org/architecture/2005/`. This attribute is specified in the topic and map modules, and it uses a default value of the current version of DITA. The current default is "2.0".

@specializations (architectural attributes)

Specifies the attribute-domain specializations that are included in the document-type shell. This attribute is set as a default within the document-type shell. The value varies depending on what domains are integrated into the document-type shell. For example, a grammar file that includes the specialized attributes `@audience`, `@deliveryTarget`, and `@newBaseAtt` would set the value to `@props/audience @props/deliveryTarget @base/newBaseAtt`.

@xmlns:ditaarch (architectural attributes)

Declares the default DITA namespace. This namespace is declared as such in the RNG modules for `<topic>` and `<map>`, but it is specified as an attribute in the equivalent DTD-based modules. The value is fixed to "http://dita.oasis-open.org/architecture/2005/".

Common map attributes

This group contains attributes that are frequently used on map elements.

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> I've added draft comments to the attribute definitions in this section that explain how the attribute is defined in the "DITA map attributes" topic.
>
> **Disposition: Unassigned**

@cascade (common map attributes)

Specifies how metadata attributes cascade within a map. The specification defines the following values:

merge

Indicates that the metadata attributes cascade, and that the values of the metadata attributes are additive. This is the processing default for the @cascade attribute.

nomerge

Indicates that the metadata attributes cascade, but that they are not additive for <topicref> elements that specify a different value for a specific metadata attribute. If the cascading value for an attribute is already merged based on multiple ancestor elements, that merged value continues to cascade until a new value is encountered. That is, setting cascade="nomerge" does not undo merging that took place on ancestor elements.

Processors can also define custom, implementation-specific tokens for this attribute.

See Cascading of metadata attributes in a DITA map for more information about how this attribute interacts with metadata attributes.

@chunk (common map attributes)

Specifies how a processor should render a map or branch of a map. For example, it can be used to specify that individual topic documents should be rendered as a single document, or that a single document with multiple topics should be rendered as multiple documents.

The following values are valid:

combine

Instructs a processor to combine the referenced source documents for rendering purposes. This is intended for cases where a publishing process normally results in a single output artifact for each source XML document.

split

Instructs a processor to split each topic from the referenced source document into its own document for rendering purposes. This is intended for cases where a publishing process normally results in a single output artifact for each source XML document, regardless of how many DITA topics exist within each source document.

Processors can also define custom, implementation-specific tokens for this attribute.

For a detailed description of the @chunk attribute and its usage, see Chunking.

@collection-type (common map attributes)

Specifies how topics or links relate to each other. The processing default is "unordered", although no default is specified in the OASIS-provided grammar files. The following values are valid:

unordered

Indicates that the order of the child topics is not significant.

sequence

Indicates that the order of the child topics is significant. Output processors will typically link between them in order.

choice

Indicates that one of the children should be selected.

family

Indicates a tight grouping in which each of the referenced topics not only relates to the current topic but also relate to each other.

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> Here is the content from the "DITA map attributes" topic:

@collection-type

The @collection-type attribute specifies how the children of a <topicref> element relate to their parent and to each other. This attribute, which is set on the parent element, typically is used by processors to determine how to generate navigation links in the rendered topics. For example, a @collection-type value of "sequence" indicates that children of the specifying <topicref> element represent an ordered sequence of topics; processors might add numbers to the list of child topics or generate next/previous links for online presentation. This attribute is available in topics on the <linklist> and <linkpool> elements, where it has the same behavior. Where the @collection-type attribute is available on elements that cannot directly contain elements, the behavior of the attribute is undefined.

Disposition: Unassigned

Comment by Kristen J Eberlein on 28 September 2022

In the definitions of the supported values, do we want to refer to "resources" instead of "topics"? Since we specify that @collection-type specifies "how topics **or links** relate to each other" ...

Disposition: Unassigned

@keyscope (common map attributes)

Specifies that the element marks the boundaries of a key scope.

See B.4 STUB CONTENT (58) for information on using this attribute.

Comment by Kristen J Eberlein on 28 September 2022

Here is the content from the "DITA map attributes" topic:

@keyscope

Defines a new scope for key definition and resolution, and gives the scope one or more names. For more information about key scopes, see Indirect key-based addressing.

Disposition: Unassigned

@linking (common map attributes)

Specifies linking characteristics of a topic specific to the location of this reference in a map. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see Cascading of metadata attributes in a DITA map).

Comment by robander on Dec 28 2021

The text below matches 1.3 spec text but I'm nervous about "cannot link" type definition. It's describing how to generate links based on the current context in the map - it's not describing what the topic itself is allowed to link to, which is how I interpret "can".

Disposition: Unassigned

The following values are valid:

targetonly

A topic can only be linked to and cannot link to other topics.

sourceonly

A topic cannot be linked to but can link to other topics.

normal
A topic can be linked to and can link to other topics. Use this to override the linking value of a parent topic.

none
A topic cannot be linked to or link to other topics.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

Comment by Kristen J Eberlein on 28 September 2022

Here is the content from the "DITA map attributes" topic:

@linking

By default, the relationships between the topics that are referenced in a map are reciprocal:

- Child topics link to parent topics and vice versa.
- Next and previous topics in a sequence link to each other.
- Topics in a family link to their sibling topics.
- Topics referenced in the table cells of the same row in a relationship table link to each other. A topic referenced within a table cell does not (by default) link to other topics referenced in the same table cell.

This behavior can be modified by using the `@linking` attribute, which enables an author or information architect to specify how a topic participates in a relationship. The following values are valid:

linking="none"
Specifies that the topic does not exist in the map for the purposes of calculating links.

linking="sourceonly"
Specifies that the topic will link to its related topics but not vice versa.

linking="targetonly"
Specifies that the related topics will link to it but not vice versa.

linking="normal"
Default value. It specifies that linking will be reciprocal (the topic will link to related topics, and they will link back to it).

Authors also can create links directly in a topic by using the `<xref>` or `<link>` elements, but in most cases map-based linking is preferable, because links in topics create dependencies between topics that can hinder reuse.

Note that while the relationships between the topics that are referenced in a map are reciprocal, the relationships merely *imply* reciprocal links in generated output that includes links. The rendered navigation links are a function of the presentation style that is determined by the processor.

Disposition: Unassigned

@processing-role (common map attributes)
Specifies whether the referenced resource is processed normally or treated as a resource that is only included in order to resolve references, such as key or content references. The following values are valid:

normal

Indicates that the resource is a readable part of the information set. It is included in navigation and search results. This is the default value for the `<topicref>` element.

resource-only

Indicates that the resource should be used only for processing purposes. It is not included in navigation or search results, nor is it rendered as a topic. This is the default value for the `<keydef>` element.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@search (common map attributes)

Specifies whether the target is available for searching. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see Cascading of metadata attributes in a DITA map). The following values are valid: "yes", "no", and "-dita-use-conref-target".

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> Here is the content from the "DITA map attributes" topic:
>
> **@search**
>
> Specifies whether the topic is included in search indexes.
>
> **Disposition: Unassigned**

@subjectrefs (common map attributes)

Specifies one or more keys that are each defined by a subject definition in a subject scheme map. Multiple values are separated by white space.

@toc (common map attributes)

Specifies whether a topic appears in the table of contents (TOC) based on the current map context. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see Cascading of metadata attributes in a DITA map). The following values are valid:

yes

The topic appears in a generated TOC.

no

The topic does not appear in a generated TOC.

-dita-use-conref-target

See B.4 STUB CONTENT (58) for more information.

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> Here is the content from the "DITA map attributes" topic:
>
> **@toc**
>
> Specifies whether topics are excluded from navigation output, such as a Web site map or an online table of contents. By default, `<topicref>` hierarchies are included in navigation output; relationship tables are excluded.
>
> **Disposition: Unassigned**

Complex table attributes

This group includes attributes that are defined on complex table elements. Unless other noted, these attributes are part of the OASIS Exchange Table Model. Complex table elements typically use only a subset of the attributes that are defined in this group.

@align (complex table attributes)

Specifies the horizontal alignment of text in table entries. The following values are valid:

left

Indicates left alignment of the text.

right

Indicates right alignment of the text.

center

Indicates center alignment of the text.

justify

Justifies the contents to both the left and the right.

char

Indicates character alignment. The text is aligned with the first occurrence of the character specified by the `@char` attribute.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

The `@align` attribute is available on the following table elements: `<colspec>`, `<entry>`, and `<tgroup>`.

@char (complex table attributes)

Specifies the alignment character, which is the character that is used for aligning the text in table entries. This attribute applies when `align="char"`. A value of "" (the null string) means there is no aligning character.

For example, if `align="char"` and `char="."` are specified, then text in the table entry aligns with the first occurrence of the period within the entry. This might be useful if decimal alignment is required.

The `@char` attribute is available on the following table elements: `<colspec>` and `<entry>`.

@charoff (complex table attributes)

Specifies the horizontal offset of the alignment character that is specified by the `@char` attribute. The value is a greater-than-zero number that is less than or equal to 100. It represents the percentage of the current column width by which the text is offset to the left of the alignment character.

For example, if `align="char"`, `char="."`, and `charoff="50"` are all specified, then text in the table entry is aligned 50% of the distance to the left of the first occurrence of the period character within the table entry.

The `@charoff` attribute is available on the following table elements: `<colspec>` and `<entry>`.

@colsep (complex table attributes)

Specifies whether to render column separators between table entries. The following values are valid: "0" (no separators) and "1" (separators).

The `@colsep` attribute is available on the following table elements: `<colspec>`, `<entry>`, `<table>`, and `<tgroup>`.

@rowheader (complex table attributes)

Specifies whether the entries in the respective column are row headers. The following values are valid:

firstcol

Indicates that entries in the first column of the table are row headers. This applies when the `@rowheader` attribute is specified on the `<table>` element.

headers

Indicates that entries of the column that is described using the `<colspec>` element are row headers. This applies when the `@rowheader` attribute is specified on the `<colspec>` element.

norowheader

Indicates that entries in the first column are not row headers. This applies when the `@rowheader` attribute is specified on the `<table>` element.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

Note This attribute is not part of the OASIS Exchange Table Model upon which DITA tables are based. Some processors or output formats might not support all values.

The `@rowheader` attribute is available on the following table elements: `<table>` and `<colspec>`.

@rowsep (complex table attributes)

Specifies whether to render row separators between table entries. The following values are valid: "0" (no separators) and "1" (separators).

The `@rowsep` attribute is available on the following table elements: `<colspec>`, `<entry>`, `<row>`, `<table>`, and `<tgroup>`.

@valign (complex table attributes)

Specifies the vertical alignment of text in table entries. The following values are valid:

bottom

Indicates that text is aligned with the bottom of the table entry.

middle

Indicates that text is aligned with the middle of the table entry.

top

Indicates that text is aligned with the top of the table entry.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

The `@valign` attribute is available on the following table elements: `<entry>`, `<tbody>`, `<thead>`, and `<row>`.

Data-element attributes

This group contains attributes that are defined on the `<data>` element and its specializations.

@datatype (data-element attributes)

Specifies the type of data contained in the `@value` attribute or within the `<data>` element. A typical use of `@datatype` will be the identifying URI for an XML Schema datatype.

@name (data-element attributes)

Defines a unique name for the object.

Comment by robander

Do we need to specify the scope of "unique" here?
Disposition: Unassigned

@value (data-element attributes)
Specifies a value associated with the current property or element.

Date attributes

This group contains attributes that take date values. They are defined on metadata elements that work with date information:

@expiry (date attributes)
Specifies the date when the information should be retired or refreshed. The date is specified using the ISO 8601 format: *YYYY-MM-DD*, where *YYYY* is the year, *MM* is the month (01 to 12), and *DD* is the day (01-31).

@golive (date attributes)
Specifies the publication or general availability (GA) date. The date is specified using the ISO 8601 format: *YYYY-MM-DD*, where *YYYY* is the year, *MM* is the month (01 to 12), and *DD* is the day (01-31).

Display attributes

This group contains attributes that affect the rendering of many elements.

@expanse (display attributes)
Specifies the horizontal placement of the element. The following values are valid:

column
Indicates that the element is aligned with the current column margin.

page
Indicates that the element is placed on the left page margin for left-to-right presentation or the right page margin for right-to-left presentation.

spread
Indicates that the object is rendered across a multi-page spread. If the output format does not have anything that corresponds to spreads, then "spread" has the same meaning as "page".

textline
Indicates that the element is aligned with the left (for left-to-right presentation) or right (for right-to-left presentation) margin of the current text line and takes indentation into account.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

For `<table>`, in place of the `@expanse` attribute that is used by other DITA elements, the `@pgwide` attribute is used in order to conform to the OASIS Exchange Table Model.

Some processors or output formats might not support all values.

@frame (display attributes)
Specifies which portion of a border surrounds the element. The following values are valid:

all
Indicates that a line is rendered at the top, bottom, left, and right of the containing element.

bottom
Indicates that a line is rendered at the bottom of the containing element.

none
Indicates that no lines are rendered.

sides
Indicates that a line is rendered at the left and right of the containing element.

top
Indicates that a line is rendered at the top of the containing element.

topbot
Indicates that a line is rendered at the top and bottom of the containing element.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

Some processors or output formats might not support all values.

@scale (display attributes)
Specifies the percentage by which fonts are resized in relation to the normal text size. The value of this attribute is a positive integer. When used on `<table>` or `<simpletable>`, the following values are valid: "50", "60", "70", "80", "90", "100", "110", "120", "140", "160", "180", "200", and -dita-use-conref-target (58).

This attribute is primarily useful for print-oriented display. Some processors might not support all values.

If the `@scale` attribute is specified on an element that contains an image, the image is not scaled. The image is scaled **only** if a scaling property is explicitly specified for the `<image>` element.

ID and conref attributes

This group contains the attributes that enable the naming and referencing of elements.

@conaction
Specifies how the element content will be pushed into a new location. The following values are valid:

mark
The element acts as a marker when pushing content before or after the target, to help ensure that the push action is valid. The element with `conaction="mark"` also specifies the target of the push action with `@conref`. Content inside of the element with `conaction="mark"` is not pushed to the new location.

pushafter
Content from this element is pushed after the location specified by `@conref` on the element with `conaction="mark"`. The element with `conaction="pushafter"` is the first sibling element after the element with `conaction="mark"`.

pushbefore
Content from this element is pushed before the location specified by `@conref` on the element with `conaction="mark"`. The element with `conaction="pushbefore"` is the first sibling element before the element with `conaction="mark"`.

pushreplace
Content from this element replaces any content from the element referenced by the `@conref` attribute. A second element with `conaction="mark"` is not used when using `conaction="pushreplace"`.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

See B.4 STUB CONTENT (58) for examples and details about the syntax.

@conkeyref
Specifies a key name or a key name with an element ID that acts as an indirect reference to reusable content. The referenced content is used in place of the content of the current element. See B.4 STUB CONTENT (58) for more details about the syntax and behaviors.

@conref
Specifies a URI that references a DITA element. The referenced content is used in place of the content of the current element. See B.4 STUB CONTENT (58) for examples and details about the syntax.

@conrefend
Specifies a URI that references the last element in a sequence of elements, with the first element of the sequence specified by `@conref`. The referenced sequence of elements is used in place of the content of the current element. See B.4 STUB CONTENT (58) for examples and details about the syntax.

@id
Specifies an identifier for the current element. This ID is the target for references by `@href` and `@conref` attributes and for external applications that refer to DITA or LwDITA content. This attribute is defined with the XML data type NMTOKEN, except where noted for specific elements within the language reference.

See id attribute for more details.

Inclusion attributes

This group includes attributes defined on `<include>` and its specializations:

Comment by Kristen J Eberlein on 28 September 2002

What is specialized from `<include>`? Both base (if any) and technical content ...

Disposition: Unassigned

@encoding (inclusion attributes)

Comment by Kristen J Eberlein on 29 April 2019

Can we replace "should" in the following definition?

Disposition: Unassigned

Specifies the character encoding to use when translating the character data from the referenced content. The value should be a valid encoding name. If not specified, processors may make attempts to automatically determine the correct encoding, for example using HTTP headers, through analysis of the binary structure of the referenced data, or the `<?xml?>` processing instruction when including XML as text. The resource should be treated as UTF-8 if no other encoding information can be determined.

When `parse="xml"`, standard XML parsing rules apply for the detection of character encoding. The necessity and uses of `@encoding` for non-standard values of `@parse` are implementation-dependent.

@parse (inclusion attributes)
Specifies the processing expectations for the referenced resource. Processors must support the following values:

text

The contents should be treated as plain text. Reserved XML characters should be displayed, and not interpreted as XML markup.

xml

The contents of the referenced resource should be treated as an XML document, and the referenced element should be inserted at the location of the `<include>` element. If a fragment identifier is included in the address of the content, processors must select the element with the specified ID. If no fragment identifier is included, the root element of the referenced XML document is selected. Any grammar processing should be performed during resolution, such that default attribute values are explicitly populated. Prolog content must be discarded.

It is an error to use `parse="xml"` anywhere other than within `<foreign>` or a specialization thereof.

Processors may support other values for the `@parse` attribute with proprietary processing semantics. Processors should issue warnings and use `<fallback>` when they encounter unsupported `@parse` values. Non-standard `@parse` instructions should be expressed as URIs.

Note Proprietary `@parse` values will likely limit the portability and interoperability of DITA content, so should be used with care.

Link relationship attributes

This group contains attributes whose values can be used for representing navigational relationships.

@format (link-relationship attributes)
Specifies the format of the resource that is referenced. See B.4 STUB CONTENT (58) for detailed information on supported values and processing implications.

@href (link-relationship attributes)
Specifies a reference to a resource. See B.4 STUB CONTENT (58) for detailed information on supported values and processing implications.

@scope (link-relationship attributes)
Specifies the closeness of the relationship between the current document and the referenced resource. The following values are valid: "local", "peer", "external", and "-dita-use-conref-target".

See B.4 STUB CONTENT (58) for detailed information on supported values and processing implications.

@type (link-relationship attributes)
Describes the target of a reference. See B.4 STUB CONTENT (58) for detailed information on supported values and processing implications.

Localization attributes

Comment by Kristen J Eberlein on 29 September 2022

The definition of the localizations attribute matches how they are described in the architectural topics. Wherever possible, the definition is reused. Where it is not reused (because the definition in the archSpec topics is in a shortdesc), I've checked to ensure that wording is identical.

Disposition: Unassigned

This group contains the attributes that are related to translation and localization.

@dir

Identifies or overrides the text directionality. The following values are valid:

lro
Indicates an override of the Unicode Bidirectional Algorithm, forcing the element into left-to-right mode.

ltr
Indicates left-to-right.

rlo
Indicates an override of the Unicode Bidirectional Algorithm, forcing the element into right-to-left mode.

rtl
Indicates right-to-left.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

See The dir attribute for more information.

@translate
Specifies whether the content of the element should be translated. The following values are valid: "yes", "no", and "-dita-use-conref-target".

See Element-by-element recommendations for translators for suggested processing defaults for each element.

Comment by Kristen J Eberlein on 31 December 2021

Does Element-by-element recommendations for translators really provide suggested processing defaults for each element? I thought it covered whether an element was block or in-line and whether there were considerations that translators needed to be aware of.

Disposition: Unassigned

@xml:lang
Specifies the language and optional locale of the content that is contained in an element. Valid values are language tokens or the null string. The `@xml:lang` attribute and its values are described in the Extensible Markup Language 1.0 specification, fifth edition.

Comment by Kristen J Eberlein on 29 September 2022

Do we also want to direct readers to the architectural topics about the `@xml:lang` attribute?

Disposition: Unassigned

Metadata attributes

This group contains common metadata attributes: `@base`, `@importance`, `@props`, `@rev`, and `@status`. The `@base` and `@props` attributes can be specialized.

@base
Specifies metadata about the element. It is often used as a base for specialized attributes that have a simple syntax for values, but which are not conditional processing attributes.

The @base attribute takes a space-delimited set of values. However, when serving as a container for generalized attributes, the attribute values will be more complex. See Attribute generalization for more details.

@importance

Specifies the importance or priority that is assigned to an element. The following values are valid: "default", "deprecated", "high", "low", "normal", "obsolete", "optional", "recommended", "required", "urgent", and "-dita-use-conref-target". This attribute is not used for conditional processing, although applications might use the value of the @importance attribute to highlight elements. For example, in steps of a task topic, the value of the @importance attribute indicates whether a step is optional or required.

> **Comment by Kristen J Eberlein on 29 September 2022**
>
> I think the phrase "to highlight elements" is a little off. Maybe "render generated text"? And how about adding "Processors often add text or images to ensure that readers of the generated content understand whether the step is optional or required." to the end of the example?
>
> **Disposition: Unassigned**

@props

Specifies metadata about the element. New attributes can be specialized from the @props attribute. This attribute supports conditional processing. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

The @props attribute takes a space-delimited set of values. However, when serving as a container for generalized attributes, the attribute values will be more complex. See Attribute generalization for more details.

@rev

Specifies a revision level of an element that identifies when the element was added or modified. It can be used to flag outputs when it matches a run-time parameter. It cannot be used for filtering nor is it sufficient to be used for version control. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

> **Comment by Kristen J Eberlein on 29 September 2022**
>
> I want to tweak this. How about the following? Also, neither definition describes what values are permitted.
>
> Specifies metadata that identifies when the element was added or the content of the element was modified. The @rev attribute can be used for flagging. It cannot be used for filtering nor is it sufficient to be used for version control. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.
>
> **Disposition: Unassigned**

@status

Specifies the modification status of the element. The following values are valid: "new", "changed", "deleted", "unchanged", and "-dita-use-conref-target".

Simple table attributes

This group includes attributes that are defined only on the `<simpletable>` element: `@keycol` and `@relcolwidth`. These attributes are listed in a group because the `<simpletable>` element is frequently used as a specialization base.

@keycol (simpletable attributes)

Specifies the column that contains the content that represents the key to the tabular structure. If `@keycol` is present and assigned a numerical value, the specified column is treated as a vertical header.

@relcolwidth (simpletable attributes)

Specifies the width of each column in relationship to the width of the other columns. The value is a space-separated list of relative column widths. Each column width is specified as a positive integer or decimal number followed by an asterisk character.

For example, the value `relcolwidth="1* 2* 3*"` gives a total of 6 units across three columns. The relative widths are 1/6, 2/6, and 3/6 (16.7%, 33.3%, and 50%). Similarly, the value `relcolwidth="90* 150*"` causes relative widths of 90/240 and 150/240 (37.5% and 62.5%).

Universal attributes

This group defines a set of attributes that are available on almost all DITA elements. It includes all elements in the ID, localization, and metadata attribute groups, as well as the following attributes:

@class *(not for use by authors)*

This attribute is not for use by authors. If an editor displays `@class` attribute values, do not edit them. Specifies a default value that defines the specialization ancestry of the element. Its predefined values allow DITA tools to work correctly with specialized elements. In a generalized DITA document the `@class` attribute value in the generalized instance might differ from the default value for the `@class` attribute for the element as given in the DTD or schema. See The class attribute rules and syntax for more information. This attribute is specified on every element except for the `<dita>` container element. It is always specified with a default value, which varies for each element.

@outputclass

Specifies a role that the element is playing. The role must be consistent with the basic semantic and expectations for the element. In particular, the `@outputclass` attribute can be used for styling during output processing; HTML output will typically preserve `@outputclass` for CSS processing.

> **Comment by robander**
> I don't like "The role must be consistent...", that seems like best practice that cannot be normative – and I could easily say outputclass="flashy" which makes my element show up with sparkles, and has nothing to do with "the basic semantic and expectations for the element".
> **Disposition: Unassigned**

B.2 Universal attribute group

The universal attribute group defines a set of common attributes that are available on almost every DITA element. The universal attribute group includes all attributes from the ID, localization, and metadata attribute groups, plus the `@class` and `@outputclass` attributes.

> **Comment by Kristen J Eberlein on 29 December 2021**
>
> This is something wrong with the organizational structure of this topic ... Look at it in outline form, and check that the sections, titles, and content all make logical sense with the topic title of "Universal attribute group".

Disposition: Unassigned

Common attribute groups

The following attribute groups are referenced in this specification. They are also used in the grammar files when the element attributes are defined.

Universal attributes

Includes `@class` and `@outputclass`, along with every attribute in the ID, localization, and metadata attribute groups.

ID attributes

This group includes the attributes that enable the naming and referencing of elements: `@conaction`, `@conkeyref`, `@conref`, `@conrefend`, and `@id`.

Localization attributes

This group includes attributes that are related to translation and localization: `@dir`, `@translate`, and `@xml:lang`.

Metadata attributes

Comment by Kristen J Eberlein on 31 December 2021

Why do we need to mention that two attributes are available for specialization here? I think it makes the paragraph hard to read.

Disposition: Unassigned

This group includes common metadata attributes, two of which are available for specialization: `@base`, `@importance`, `@props`, `@rev`, and `@status`.

The base DITA vocabulary from OASIS includes several specializations of `@props`: `@audience`, `@deliveryTarget`, `@otherprops`, `@platform`, and `@product`. These attributes are defined as attribute-extension domains. By default, they are integrated into all OASIS-provided document-type shells, but they can be made unavailable by implementing custom document-type shells.

Comment by Kristen J Eberlein on 29 December 2021

Why do we provide information about specialization and custom document-type shells here? I think that information could be removed.

Disposition: Unassigned

Universal attribute definitions

The universal attributes for OASIS DITA elements are defined below. Specialized attributes, which are part of the OASIS distribution but are only available when explicitly included in a shell, are noted in the list.

@audience *(specialized attribute)*

Indicates the intended audience for the element. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@base

Specifies metadata about the element. It is often used as a base for specialized attributes that have a simple syntax for values, but which are not conditional processing attributes.

The `@base` attribute takes a space-delimited set of values. However, when serving as a container for generalized attributes, the attribute values will be more complex. See Attribute generalization for more details.

@class *(not for use by authors)*

This attribute is not for use by authors. If an editor displays `@class` attribute values, do not edit them. Specifies a default value that defines the specialization ancestry of the element. Its predefined values allow DITA tools to work correctly with specialized elements. In a generalized DITA document the `@class` attribute value in the generalized instance might differ from the default value for the `@class` attribute for the element as given in the DTD or schema. See The class attribute rules and syntax for more information. This attribute is specified on every element except for the `<dita>` container element. It is always specified with a default value, which varies for each element.

@conaction

Specifies how the element content will be pushed into a new location. The following values are valid:

mark

The element acts as a marker when pushing content before or after the target, to help ensure that the push action is valid. The element with `conaction="mark"` also specifies the target of the push action with `@conref`. Content inside of the element with `conaction="mark"` is not pushed to the new location.

pushafter

Content from this element is pushed after the location specified by `@conref` on the element with `conaction="mark"`. The element with `conaction="pushafter"` is the first sibling element after the element with `conaction="mark"`.

pushbefore

Content from this element is pushed before the location specified by `@conref` on the element with `conaction="mark"`. The element with `conaction="pushbefore"` is the first sibling element before the element with `conaction="mark"`.

pushreplace

Content from this element replaces any content from the element referenced by the `@conref` attribute. A second element with `conaction="mark"` is not used when using `conaction="pushreplace"`.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

See B.4 STUB CONTENT (58) for examples and details about the syntax.

@conkeyref

Specifies a key name or a key name with an element ID that acts as an indirect reference to reusable content. The referenced content is used in place of the content of the current element. See B.4 STUB CONTENT (58) for more details about the syntax and behaviors.

@conref

Specifies a URI that references a DITA element. The referenced content is used in place of the content of the current element. See B.4 STUB CONTENT (58) for examples and details about the syntax.

@conrefend

Specifies a URI that references the last element in a sequence of elements, with the first element of the sequence specified by `@conref`. The referenced sequence of elements is used in place of the content of the current element. See B.4 STUB CONTENT (58) for examples and details about the syntax.

@deliveryTarget *(specialized attribute)*

Specifies the intended delivery target of the content, for example, "html", "pdf", or "epub". If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@dir

Identifies or overrides the text directionality. The following values are valid:

lro

Indicates an override of the Unicode Bidirectional Algorithm, forcing the element into left-to-right mode.

ltr

Indicates left-to-right.

rlo

Indicates an override of the Unicode Bidirectional Algorithm, forcing the element into right-to-left mode.

rtl

Indicates right-to-left.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

See The dir attribute for more information.

@id

Specifies an identifier for the current element. This ID is the target for references by `@href` and `@conref` attributes and for external applications that refer to DITA or LwDITA content. This attribute is defined with the XML data type NMTOKEN, except where noted for specific elements within the language reference.

See id attribute for more details.

@importance

Specifies the importance or priority that is assigned to an element. The following values are valid: "default", "deprecated", "high", "low", "normal", "obsolete", "optional", "recommended", "required", "urgent", and "-dita-use-conref-target". This attribute is not used for conditional processing, although applications might use the value of the `@importance` attribute to highlight elements. For example, in steps of a task topic, the value of the `@importance` attribute indicates whether a step is optional or required.

Comment by Kristen J Eberlein on 29 September 2022

I think the phrase "to highlight elements" is a little off. Maybe "render generated text"? And how about adding "Processors often add text or images to ensure that readers of the generated content understand whether the step is optional or required." to the end of the example?

Disposition: Unassigned

@otherprops *(specialized attribute)*

Specifies a property or properties that provide selection criteria for the element. Alternatively, the `@props` attribute can be specialized to provide a new metadata attribute instead of using the general `@otherprops` attribute. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@outputclass

Specifies a role that the element is playing. The role must be consistent with the basic semantic and expectations for the element. In particular, the `@outputclass` attribute can be used for styling during output processing; HTML output will typically preserve `@outputclass` for CSS processing.

> **Comment by robander**
> I don't like "The role must be consistent...", that seems like best practice that cannot be normative – and I could easily say outputclass="flashy" which makes my element show up with sparkles, and has nothing to do with "the basic semantic and expectations for the element".
> **Disposition: Unassigned**

@platform *(specialized attribute)*

Indicates operating system and hardware. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

> **Comment by robander**
> I think this could specify a platform that is not an operating system or hardware, right? The current definition explicitly limits platform to those two … maybe "Specifies a platform or platforms to which the element applies, such as the operating system or hardware relevant to a task."
> **Disposition: Unassigned**

@product *(specialized attribute)*

Specifies the name of the product to which the element applies. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@props

Specifies metadata about the element. New attributes can be specialized from the `@props` attribute. This attribute supports conditional processing. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

The `@props` attribute takes a space-delimited set of values. However, when serving as a container for generalized attributes, the attribute values will be more complex. See Attribute generalization for more details.

@rev

Specifies a revision level of an element that identifies when the element was added or modified. It can be used to flag outputs when it matches a run-time parameter. It cannot be used for filtering nor is it sufficient to be used for version control. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

> **Comment by Kristen J Eberlein on 29 September 2022**
>
> I want to tweak this. How about the following? Also, neither definition describes what values are permitted.
>
> Specifies metadata that identifies when the element was added or the content of the element was modified. The `@rev` attribute can be used for flagging. It cannot be used for filtering nor is it sufficient to be used for version control. If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.
>
> **Disposition: Unassigned**

@status

Specifies the modification status of the element. The following values are valid: "new", "changed", "deleted", "unchanged", and "-dita-use-conref-target".

@translate

Specifies whether the content of the element should be translated. The following values are valid: "yes", "no", and "-dita-use-conref-target".

See Element-by-element recommendations for translators for suggested processing defaults for each element.

> **Comment by Kristen J Eberlein on 31 December 2021**
>
> Does Element-by-element recommendations for translators really provide suggested processing defaults for each element? I thought it covered whether an element was block or in-line and whether there were considerations that translators needed to be aware of.
>
> **Disposition: Unassigned**

@xml:lang

Specifies the language and optional locale of the content that is contained in an element. Valid values are language tokens or the null string. The `@xml:lang` attribute and its values are described in the Extensible Markup Language 1.0 specification, fifth edition.

> **Comment by Kristen J Eberlein on 29 September 2022**
>
> Do we also want to direct readers to the architectural topics about the `@xml:lang` attribute?
>
> **Disposition: Unassigned**

B.3 Common attributes

The common attributes topic collects defines most of the attributes that are used on more than one base element.

Common attribute groups

The following groups are referenced in this specification, and they are also used in grammar files when defining attributes for elements.

Architectural attributes

This group includes a set of attributes that are defined for document-level elements such as `<topic>` and `<map>`: `@DITAArchVersion`, `@specializations`, and `@xmlns:ditaarch`.

Common map attributes

This group includes attributes that are frequently used on map elements: `@cascade`, `@chunk`, `@collection-type`, `@keyscope`, `@linking`, `@processing-role`, `@search`, `@toc`, and `@subjectrefs`.

Complex table attributes

This group includes attributes that are defined on table elements but not simple table elements. These attributes are part of the OASIS Exchange Table Model, unless otherwise noted. Table elements generally use only a subset of the attributes that are defined in this group. This group contains the following attributes: `@align`, `@char`, `@charoff`, `@colsep`, `@rowheader`, `@rowsep`, and `@valign`.

Data-element attributes

Includes attributes defined on `<data>` and its many specializations: `@datatype`, `@name`, and `@value`

Date attributes

Includes attributes that take date values, and are defined on metadata elements that work with date information: `@expiry` and `@golive`

Display attributes

This group includes attributes that affect the rendering of many elements: `@expanse`, `@frame`, and `@scale`.

Inclusion attributes

Includes attributes defined on `<include>` and its specializations: `@encoding` and `@parse`.

Link-relationship attributes

This group includes attributes whose values can be used for representing navigational relationships: `@format`, `@href`, `@type`, and `@scope`.

Simple table attributes

> **Comment by Kristen J Eberlein on 29 December 2021**
>
> If I have jumped to this place in a document from the element-reference topic, I want the attributes listed here in the "Simple table group" to be hyperlinked to the actual definition.
>
> **Disposition: Unassigned**

This group includes attributes that are defined only on the `<simpletable>` element: `@keycol` and `@relcolwidth`. These attributes are listed in a group because the `<simpletable>` element is frequently used as a specialization base.

Table accessibility attributes

This group contains attributes that are defined on the `<stentry>` element and its specializations: `@headers` (52) and `@scope` (as defined on `<stentry>`) (56).

Other attributes (not in a group)

These are attributes that are used in the same way on more than one base element, but they are not formally grouped together: `@compact`, `@duplicates`, `@otherrole`, `@role`, and `@title-role`.

Common attribute definitions

Common attributes, including those in the groups listed above, are defined as follows.

@align (complex table attributes)

Specifies the horizontal alignment of text in table entries. The following values are valid:

left

Indicates left alignment of the text.

right

Indicates right alignment of the text.

center

Indicates center alignment of the text.

justify

Justifies the contents to both the left and the right.

char
: Indicates character alignment. The text is aligned with the first occurrence of the character specified by the `@char` attribute.

-dita-use-conref-target
: See Using the -dita-use-conref-target value for more information.

The `@align` attribute is available on the following table elements: `<colspec>`, `<entry>`, and `<tgroup>`.

@cascade (common map attributes)

Specifies how metadata attributes cascade within a map. The specification defines the following values:

merge
: Indicates that the metadata attributes cascade, and that the values of the metadata attributes are additive. This is the processing default for the `@cascade` attribute.

nomerge
: Indicates that the metadata attributes cascade, but that they are not additive for `<topicref>` elements that specify a different value for a specific metadata attribute. If the cascading value for an attribute is already merged based on multiple ancestor elements, that merged value continues to cascade until a new value is encountered. That is, setting `cascade="nomerge"` does not undo merging that took place on ancestor elements.

Processors can also define custom, implementation-specific tokens for this attribute.

See Cascading of metadata attributes in a DITA map for more information about how this attribute interacts with metadata attributes.

@char (complex table attributes)

Specifies the alignment character, which is the character that is used for aligning the text in table entries. This attribute applies when `align="char"`. A value of "" (the null string) means there is no aligning character.

For example, if `align="char"` and `char="."` are specified, then text in the table entry aligns with the first occurrence of the period within the entry. This might be useful if decimal alignment is required.

The `@char` attribute is available on the following table elements: `<colspec>` and `<entry>`.

@charoff (complex table attributes)

Specifies the horizontal offset of the alignment character that is specified by the `@char` attribute. The value is a greater-than-zero number that is less than or equal to 100. It represents the percentage of the current column width by which the text is offset to the left of the alignment character.

For example, if `align="char"`, `char="."`, and `charoff="50"` are all specified, then text in the table entry is aligned 50% of the distance to the left of the first occurrence of the period character within the table entry.

The `@charoff` attribute is available on the following table elements: `<colspec>` and `<entry>`.

@chunk (common map attributes)

Specifies how a processor should render a map or branch of a map. For example, it can be used to specify that individual topic documents should be rendered as a single document, or that a single document with multiple topics should be rendered as multiple documents.

The following values are valid:

combine

Instructs a processor to combine the referenced source documents for rendering purposes. This is intended for cases where a publishing process normally results in a single output artifact for each source XML document.

split

Instructs a processor to split each topic from the referenced source document into its own document for rendering purposes. This is intended for cases where a publishing process normally results in a single output artifact for each source XML document, regardless of how many DITA topics exist within each source document.

Processors can also define custom, implementation-specific tokens for this attribute.

For a detailed description of the `@chunk` attribute and its usage, see Chunking.

@collection-type (common map attributes)

Specifies how topics or links relate to each other. The processing default is "unordered", although no default is specified in the OASIS-provided grammar files. The following values are valid:

unordered

Indicates that the order of the child topics is not significant.

sequence

Indicates that the order of the child topics is significant. Output processors will typically link between them in order.

choice

Indicates that one of the children should be selected.

family

Indicates a tight grouping in which each of the referenced topics not only relates to the current topic but also relate to each other.

Comment by Kristen J Eberlein on 28 September 2022

Here is the content from the "DITA map attributes" topic:

@collection-type

The `@collection-type` attribute specifies how the children of a `<topicref>` element relate to their parent and to each other. This attribute, which is set on the parent element, typically is used by processors to determine how to generate navigation links in the rendered topics. For example, a `@collection-type` value of "sequence" indicates that children of the specifying `<topicref>` element represent an ordered sequence of topics; processors might add numbers to the list of child topics or generate next/previous links for online presentation. This attribute is available in topics on the `<linklist>` and `<linkpool>` elements, where it has the same behavior. Where the `@collection-type` attribute is available on elements that cannot directly contain elements, the behavior of the attribute is undefined.

Disposition: Unassigned

Comment by Kristen J Eberlein on 28 September 2022

In the definitions of the supported values, do we want to refer to "resources" instead of "topics"? Since we specify that `@collection-type` specifies "how topics **or links** relate to each other" ...

Disposition: Unassigned

@colsep (complex table attributes)

Specifies whether to render column separators between table entries. The following values are valid: "0" (no separators) and "1" (separators).

The @colsep attribute is available on the following table elements: <colspec>, <entry>, <table>, and <tgroup>.

@compact

Specifies whether the vertical spacing between list items is tightened. The following values are valid: "yes", "no", and "-dita-use-conref-target". Some DITA processors or output formats might not support the @compact attribute.

@datatype (data-element attributes)

Specifies the type of data contained in the @value attribute or within the <data> element. A typical use of @datatype will be the identifying URI for an XML Schema datatype.

@DITAArchVersion (architectural attributes)

Specifies the version of the DITA architecture that is in use. This attribute is in the namespace `http://dita.oasis-open.org/architecture/2005/`. This attribute is specified in the topic and map modules, and it uses a default value of the current version of DITA. The current default is "2.0".

@duplicates

Specifies whether duplicate links are removed from a group of links. Duplicate links are links that address the same resource using the same properties, such as link text and link role. How duplicate links are determined is processor-specific. The following values are valid:

yes

Specifies that duplicate links are retained.

no

Specifies that duplicate links are removed.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

The suggested processing default is "yes" within <linklist> elements and "no" for other links.

> **Comment by robander on Dec 28 2021**
> "How duplicate links are determined is processor-specific" ==> this should be included in any updates to standardize language around "implementation dependent".
> **Disposition: Unassigned**

@encoding (inclusion attributes)

> **Comment by Kristen J Eberlein on 29 April 2019**
>
> Can we replace "should" in the following definition?
>
> **Disposition: Unassigned**

Specifies the character encoding to use when translating the character data from the referenced content. The value should be a valid encoding name. If not specified, processors may make attempts to automatically determine the correct encoding, for example using HTTP headers, through analysis of the binary structure of the referenced data, or the <?xml?> processing instruction when including XML as text. The resource should be treated as UTF-8 if no other encoding information can be determined.

When `parse="xml"`, standard XML parsing rules apply for the detection of character encoding. The necessity and uses of `@encoding` for non-standard values of `@parse` are implementation-dependent.

@expanse (display attributes)

Specifies the horizontal placement of the element. The following values are valid:

column

Indicates that the element is aligned with the current column margin.

page

Indicates that the element is placed on the left page margin for left-to-right presentation or the right page margin for right-to-left presentation.

spread

Indicates that the object is rendered across a multi-page spread. If the output format does not have anything that corresponds to spreads, then "spread" has the same meaning as "page".

textline

Indicates that the element is aligned with the left (for left-to-right presentation) or right (for right-to-left presentation) margin of the current text line and takes indentation into account.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

For `<table>`, in place of the `@expanse` attribute that is used by other DITA elements, the `@pgwide` attribute is used in order to conform to the OASIS Exchange Table Model.

Some processors or output formats might not support all values.

@expiry (date attributes)

Specifies the date when the information should be retired or refreshed. The date is specified using the ISO 8601 format: *YYYY-MM-DD*, where *YYYY* is the year, *MM* is the month (01 to 12), and *DD* is the day (01-31).

@format (link-relationship attributes)

Specifies the format of the resource that is referenced. See B.4 STUB CONTENT (58) for detailed information on supported values and processing implications.

@frame (display attributes)

Specifies which portion of a border surrounds the element. The following values are valid:

all

Indicates that a line is rendered at the top, bottom, left, and right of the containing element.

bottom

Indicates that a line is rendered at the bottom of the containing element.

none

Indicates that no lines are rendered.

sides

Indicates that a line is rendered at the left and right of the containing element.

top

Indicates that a line is rendered at the top of the containing element.

topbot

Indicates that a line is rendered at the top and bottom of the containing element.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

Some processors or output formats might not support all values.

@golive (date attributes)

Specifies the publication or general availability (GA) date. The date is specified using the ISO 8601 format: *YYYY-MM-DD*, where *YYYY* is the year, *MM* is the month (01 to 12), and *DD* is the day (01-31).

@headers

Specifies which entries in the current table provide headers for this cell. The `@headers` attribute contains an unordered set of unique, space-separated tokens, each of which is an ID reference of an entry from the same table.

@href (link-relationship attributes)

Specifies a reference to a resource. See B.4 STUB CONTENT (58) for detailed information on supported values and processing implications.

@keycol (simpletable attributes)

Specifies the column that contains the content that represents the key to the tabular structure. If `@keycol` is present and assigned a numerical value, the specified column is treated as a vertical header.

@keyref

Specifies a key name that acts as a redirectable reference based on a key definition within a map. See B.4 STUB CONTENT (58) for information on using this attribute.

For HDITA, the equivalent of `@keyref` is `@data-keyref`

Comment by robander
The definiton above for @keyref should be synchronized with the definition in the linked section on keys.
Disposition: Unassigned

@keys

Specifies one or more names for a resource. See B.4 STUB CONTENT (58) for information on using this attribute.

For HDITA, the equivalent of `@keys` is `@data-keys`

@keyscope (common map attributes)

Specifies that the element marks the boundaries of a key scope.

See B.4 STUB CONTENT (58) for information on using this attribute.

Comment by Kristen J Eberlein on 28 September 2022

Here is the content from the "DITA map attributes" topic:

@keyscope

Defines a new scope for key definition and resolution, and gives the scope one or more names. For more information about key scopes, see Indirect key-based addressing.

Disposition: Unassigned

@linking (common map attributes)

Specifies linking characteristics of a topic specific to the location of this reference in a map. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see Cascading of metadata attributes in a DITA map).

Comment by robander on Dec 28 2021

The text below matches 1.3 spec text but I'm nervous about "cannot link" type definition. It's describing how to generate links based on the current context in the map - it's not describing what the topic itself is allowed to link to, which is how I interpret "can".
Disposition: Unassigned

The following values are valid:

targetonly
A topic can only be linked to and cannot link to other topics.

sourceonly
A topic cannot be linked to but can link to other topics.

normal
A topic can be linked to and can link to other topics. Use this to override the linking value of a parent topic.

none
A topic cannot be linked to or link to other topics.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

Comment by Kristen J Eberlein on 28 September 2022

Here is the content from the "DITA map attributes" topic:

@linking

By default, the relationships between the topics that are referenced in a map are reciprocal:

- Child topics link to parent topics and vice versa.
- Next and previous topics in a sequence link to each other.
- Topics in a family link to their sibling topics.
- Topics referenced in the table cells of the same row in a relationship table link to each other. A topic referenced within a table cell does not (by default) link to other topics referenced in the same table cell.

This behavior can be modified by using the `@linking` attribute, which enables an author or information architect to specify how a topic participates in a relationship. The following values are valid:

linking="none"
Specifies that the topic does not exist in the map for the purposes of calculating links.

linking="sourceonly"
Specifies that the topic will link to its related topics but not vice versa.

linking="targetonly"
Specifies that the related topics will link to it but not vice versa.

linking="normal"
Default value. It specifies that linking will be reciprocal (the topic will link to related topics, and they will link back to it).

Authors also can create links directly in a topic by using the `<xref>` or `<link>` elements, but in most cases map-based linking is preferable, because links in topics create dependencies between topics that can hinder reuse.

Note that while the relationships between the topics that are referenced in a map are reciprocal, the relationships merely *imply* reciprocal links in generated output that includes links. The rendered navigation links are a function of the presentation style that is determined by the processor.

Disposition: Unassigned

@name (data-element attributes)

Defines a unique name for the object.

Comment by robander
Do we need to specify the scope of "unique" here?
Disposition: Unassigned

@otherrole

Specifies an alternate role for a link relationship when the `@role` attribute is set to "other".

@parse (inclusion attributes)

Specifies the processing expectations for the referenced resource. Processors must support the following values:

text

The contents should be treated as plain text. Reserved XML characters should be displayed, and not interpreted as XML markup.

xml

The contents of the referenced resource should be treated as an XML document, and the referenced element should be inserted at the location of the `<include>` element. If a fragment identifier is included in the address of the content, processors must select the element with the specified ID. If no fragment identifier is included, the root element of the referenced XML document is selected. Any grammar processing should be performed during resolution, such that default attribute values are explicitly populated. Prolog content must be discarded.

It is an error to use `parse="xml"` anywhere other than within `<foreign>` or a specialization thereof.

Processors may support other values for the `@parse` attribute with proprietary processing semantics. Processors should issue warnings and use `<fallback>` when they encounter unsupported `@parse` values. Non-standard `@parse` instructions should be expressed as URIs.

Note Proprietary `@parse` values will likely limit the portability and interoperability of DITA content, so should be used with care.

@processing-role (common map attributes)

Specifies whether the referenced resource is processed normally or treated as a resource that is only included in order to resolve references, such as key or content references. The following values are valid:

normal

Indicates that the resource is a readable part of the information set. It is included in navigation and search results. This is the default value for the `<topicref>` element.

resource-only

Indicates that the resource should be used only for processing purposes. It is not included in navigation or search results, nor is it rendered as a topic. This is the default value for the `<keydef>` element.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

If no value is specified but the attribute is specified on a containing element within a map or within the related-links section, the value cascades from the closest containing element.

@relcolwidth (simpletable attributes)

Specifies the width of each column in relationship to the width of the other columns. The value is a space-separated list of relative column widths. Each column width is specified as a positive integer or decimal number followed by an asterisk character.

For example, the value `relcolwidth="1* 2* 3*"` gives a total of 6 units across three columns. The relative widths are 1/6, 2/6, and 3/6 (16.7%, 33.3%, and 50%). Similarly, the value `relcolwidth="90* 150*"` causes relative widths of 90/240 and 150/240 (37.5% and 62.5%).

@role

Specifies the role that a linked topic plays in relationship with the current topic.

For example, in a parent/child relationship, the role would be "parent" when the target is the parent of the current topic, and "child" when the target is the child of the current topic. This can be used to sort and classify links when rendering.

The following values are valid:

ancestor

Indicates a link to a topic above the parent topic.

child

Indicates a link to a direct child such as a directly nested or dependent topic.

cousin

Indicates a link to another topic in the same hierarchy that is not a parent, child, sibling, next, or previous.

descendant

Indicates a link to a topic below a child topic.

friend

Indicates a link to a similar topic that is not necessarily part of the same hierarchy.

next

Indicates a link to the next topic in a sequence.

other

Indicates any other kind of relationship or role. The type of role is specified as the value for the `@otherrole` attribute.

parent

Indicates a link to a topic that is a parent of the current topic.

previous

Indicates a link to the previous topic in a sequence.

sibling

Indicates a link between two children of the same parent topic.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

@rowheader (complex table attributes)

Specifies whether the entries in the respective column are row headers. The following values are valid:

firstcol
Indicates that entries in the first column of the table are row headers. This applies when the `@rowheader` attribute is specified on the `<table>` element.

headers
Indicates that entries of the column that is described using the `<colspec>` element are row headers. This applies when the `@rowheader` attribute is specified on the `<colspec>` element.

norowheader
Indicates that entries in the first column are not row headers. This applies when the `@rowheader` attribute is specified on the `<table>` element.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

Note This attribute is not part of the OASIS Exchange Table Model upon which DITA tables are based. Some processors or output formats might not support all values.

The `@rowheader` attribute is available on the following table elements: `<table>` and `<colspec>`.

@rowsep (complex table attributes)
Specifies whether to render row separators between table entries. The following values are valid: "0" (no separators) and "1" (separators).

The `@rowsep` attribute is available on the following table elements: `<colspec>`, `<entry>`, `<row>`, `<table>`, and `<tgroup>`.

@scale (display attributes)
Specifies the percentage by which fonts are resized in relation to the normal text size. The value of this attribute is a positive integer. When used on `<table>` or `<simpletable>`, the following values are valid: "50", "60", "70", "80", "90", "100", "110", "120", "140", "160", "180", "200", and -dita-use-conref-target (58).

This attribute is primarily useful for print-oriented display. Some processors might not support all values.

If the `@scale` attribute is specified on an element that contains an image, the image is not scaled. The image is scaled **only** if a scaling property is explicitly specified for the `<image>` element.

@scope (link-relationship attributes)
Specifies the closeness of the relationship between the current document and the referenced resource. The following values are valid: "local", "peer", "external", and "-dita-use-conref-target".

See B.4 STUB CONTENT (58) for detailed information on supported values and processing implications.

@scope
Specifies that the current entry is a header for other table entries. The following values are valid:

col
Indicates that the current entry is a header for all cells in the column.

colgroup
Indicates that the current entry is a header for all cells in the columns that are spanned by this entry.

row
Indicates that the current entry is a header for all cells in the row.

rowgroup
Indicates that the current entry is a header for all cells in the rows that are spanned by this entry.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

@search (common map attributes)

Specifies whether the target is available for searching. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see Cascading of metadata attributes in a DITA map). The following values are valid: "yes", "no", and "-dita-use-conref-target".

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> Here is the content from the "DITA map attributes" topic:
>
> **@search**
>
> Specifies whether the topic is included in search indexes.
>
> **Disposition: Unassigned**

@specializations (architectural attributes)

Specifies the attribute-domain specializations that are included in the document-type shell. This attribute is set as a default within the document-type shell. The value varies depending on what domains are integrated into the document-type shell. For example, a grammar file that includes the specialized attributes `@audience`, `@deliveryTarget`, and `@newBaseAtt` would set the value to `@props/audience @props/deliveryTarget @base/newBaseAtt`.

@subjectrefs (common map attributes)

Specifies one or more keys that are each defined by a subject definition in a subject scheme map. Multiple values are separated by white space.

@title-role (REQUIRED)

Specifies the role that the alternative title serves. Multiple roles are separated by white space. The following roles are defined in the specification: "linking", "navigation", "search", "subtitle", and "hint".

Processors can define custom values for the `@title-role` attribute.

@toc (common map attributes)

Specifies whether a topic appears in the table of contents (TOC) based on the current map context. If the value is not specified locally, the value might cascade from another element in the map (for cascade rules, see Cascading of metadata attributes in a DITA map). The following values are valid:

yes

The topic appears in a generated TOC.

no

The topic does not appear in a generated TOC.

-dita-use-conref-target

See B.4 STUB CONTENT (58) for more information.

> **Comment by Kristen J Eberlein on 28 September 2022**
>
> Here is the content from the "DITA map attributes" topic:
>
> **@toc**
>
> Specifies whether topics are excluded from navigation output, such as a Web site map or an online table of contents. By default, `<topicref>` hierarchies are included in navigation output; relationship tables are excluded.
>
> **Disposition: Unassigned**

@type (link-relationship attributes)

Describes the target of a reference. See B.4 STUB CONTENT (58) for detailed information on supported values and processing implications.

@value (data-element attributes)

Specifies a value associated with the current property or element.

@valign (complex table attributes)

Specifies the vertical alignment of text in table entries. The following values are valid:

bottom

Indicates that text is aligned with the bottom of the table entry.

middle

Indicates that text is aligned with the middle of the table entry.

top

Indicates that text is aligned with the top of the table entry.

-dita-use-conref-target

See Using the -dita-use-conref-target value for more information.

The `@valign` attribute is available on the following table elements: `<entry>`, `<tbody>`, `<thead>`, and `<row>`.

@xml:space

Specifies how to handle white space in the current element. This attribute is provided on `<pre>`, `<lines>`, and on elements specialized from those. It ensures that parsers respect white space that is part of the data in those elements, including line-end characters. When defined, it has a fixed value of "preserve", making it a default property of the element that cannot be changed or deleted by authors.

@xmlns:ditaarch (architectural attributes)

Declares the default DITA namespace. This namespace is declared as such in the RNG modules for `<topic>` and `<map>`, but it is specified as an attribute in the equivalent DTD-based modules. The value is fixed to "http://dita.oasis-open.org/architecture/2005/".

B.4 STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT
STUB CONTENT

STUB CONTENT
STUB CONTENT

STUB CONTENT
STUB CONTENT

STUB CONTENT
STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT (59)

STUB CONTENT (59)

```
STUB CONTENT
```

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

STUB CONTENT

Index