
Stage two: #316 Add <diagnostics> to troubleshooting topic

Add a new optional <diagnostics> element to the <troubleshooting> topic to provide a distinct, semantic location to provide guidelines for determining what is the cause of the given condition in order to then apply the appropriate remedy for that cause.

Champion

Dawn Stevens, Comtech Services, Inc.

Tracking information

Event	Date	Links
Initial suggestion	Not logged	
Stage 1 proposal accepted	Not logged	GitHub issue
Stage 2 proposal submitted to reviewers	28 April 2020	Kris Eberlein, Eberlein Consulting LLC Nancy Harrison, Individual member Scott Hudson, Boeing
Stage 2 proposal submitted	30 June 2020	E-mail
Stage 2 proposal discussed	30 June 2020	Minutes
Stage 2 proposal approved	07 June 2021	Minutes
[▶] Stage 2 and 3 proposal reopened	13 July 2021 The proposal was reopened to change the content model of <diagnostics>. Instead of containing either <diagnostics-general> or <diagnostics-steps>, <diagnostics> will also allow a single instance of both elements. This change was requested by Ericsson.	Minutes [◀]
[▶] Revised stage 2 proposal submitted to reviewer		Not applicable [◀]
[▶] Revised stage 2 proposal submitted to TC	29 July 2021	[◀]
[▶] Revised stage 2 proposal approved		[◀]

Original requirement or use case

In troubleshooting topics, there are several possible relationships between a problem's cause and its solution(s):

- One cause, one solution
- One cause, many solutions
- Many causes, one solution
- Many causes, each with their own solution

In the cases where more than one solution exists, particularly the last situation, users of the troubleshooting topic must first be able to determine which cause applies to their symptoms before they can begin to implement the solution. There is no semantic element in which to place this type of diagnostic information. Current options are to place the information in one of two places:

- In the Condition section. This section is meant to simply describe the problem, not diagnose why it is happening.
- In a Cause section. Because `<cause>` is linked to `<remedy>` in a `<troubleSolution>` pairing, this solution is problematic. Authors must have a single `<troubleSolution>` with only the `<cause>` to do the diagnostics, and then have a sequence of `<troubleSolution>` pairings to give the solutions for each issue. Also diagnostics might be procedural in nature, which forces authors to either use an `` in `<cause>` or use the even more non-semantic workaround of using `<remedy>` for the diagnostic information.

Modified grammar files

The following files must be modified:

DTDs

troubleshooting.mod

RNGs

troubleshootingMod.rng

Use cases

This proposal will enable troubleshooting authors to:

Add diagnostics tables

In a diagnostic table, authors map observable symptoms related to the condition to a specific cause. In this case, the symptoms are observable without the user having to take a specific action – they can simply observe that the symptom exists. For example, a particular light is on or off.

Add step-by-step instructions for diagnosing the problem and determining the cause

Some causes may not be simply observable, but require the user to perform a series of steps to determine the cause of the problem; for example, set breakpoints, send commands, push buttons, and so on.

New terminology

Not applicable

Proposed solution

Modify the content model of `<troublebody>` to allow zero or more new `<diagnostics>` elements after `<condition>`. The `<diagnostics>` element will in turn contain one `[▶]` or both of `[◀]` two new

diagnostic elements, `<diagnostics-general>` and `<diagnostics-steps>` to address the two use cases described earlier.

Benefits

Who will benefit from this feature?

Authors writing complex troubleshooting information.

What is the expected benefit?

Authors have a separate and semantic location to place diagnostic information.

How many people probably will make use of this feature?

Most troubleshooting authors.

How much of a positive impact is expected for the users who will make use of the feature?

Significant.

Technical requirements

This proposal involves the following changes:

Adding an element

This proposal will add three elements:

- Add `<diagnostics>` element which will:
 - Be specialized from `<bodydiv>`
 - Have a content model of `[▶] one or both of: [◀]`
 - `<diagnostics-general>`
 - `<diagnostics-steps>`
- Add a `<diagnostics-general>` element which will:
 - Be specialized from `<section>`
 - Have an identical content model as `<cause>`
 - Provide a location to add a diagnostics table or `[▶] image to illustrate the steps in the diagnostic procedure [◀]`
- Add `<diagnostics-steps>` which will:
 - Be specialized from `<section>`
 - Remove `<responsible-party>` from the `<remedy>` content model, but otherwise be identical
 - Provide a location to add step-by-step diagnostic instructions

Refactoring an element

The content model of `<troublebody>` will be modified to allow zero or one `<diagnostics>` elements following `<condition>` and preceding `<troubleSolution>`

Processing impact

None

Overall usability

Improved semantics for current and future users

Backwards compatibility

This proposal addresses the following questions:

Was this change previously announced in an earlier version of DITA?

No

Removing a document type that was shipped in DITA 1.3?

No

Removing a domain that was shipped in DITA 1.3?

No

Removing a domain from a document type shell was shipped in DITA 1.3?

No

Removing or renaming an element that was shipped in DITA 1.3?

No

Removing or renaming an attribute that was shipped in DITA 1.3?

No

Changing the meaning of an element or attribute in a way that would disallow existing usage?

No

Changing a content model by removing something that was previously allowed, or by requiring something that was not?

No

Changing specialization ancestry?

No

Removing or replacing a processing feature that was defined in DITA 1.3?

No

Are element or attribute groups being renamed or shuffled?

No

Migration plan

Might any existing documents need to be migrated?

Optionally people who used a workaround and documented their diagnostic instructions using `<condition>`, `<cause>`, and/or `<remedy>` might want to migrate their content to the more semantic section; however their existing solutions would still work.

Might any existing processors or implementations need to change their expectations?

No

Might any existing specialization or constraint modules need to be migrated?

If a specialization was created for the same purpose, a company might want to adopt the new standard instead.

Costs

This proposal has a (time and effort) impact on the following groups:

Maintainers of the grammar files

The following files will need to be modified:

- DTDS:
 - `troubleshooting.mod`
- RNG:

- troubleshootingMod.rng

Editors of the DITA specification

The changes to the DITA specification will be minor to medium:

- Update Section 2.7.1.6 to discuss new structure
- Update to section 3.10.4:
 - New sections on the three new elements
 - New example in 3.10.4.1 with new elements added

Vendors of tools

As all elements are specializations of existing elements that behave the same way, no special processing needs to be added.

DITA community-at-large

The changes should not add to a perception of DITA complexity. The changes should be simple and intuitive for end users to understand.

Producing migration instructions or tools

Not applicable

Examples

This section provides examples of the new markup enabled by this proposal.

Figure 1: Adding a diagnostics table to a troubleshooting topic

The following code sample shows how a troubleshooting topic would provide a diagnostic table to help a user determine which cause and therefore which solution applies to them.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE troubleshooting PUBLIC "-//OASIS//DTD DITA 2.0 Troubleshooting//EN"
"troubleshooting.dtd">
<troubleshooting>
  <title>Car is making funny noises.</title>
  <shortdesc/>
  <prolog/>
  <troublebody>
    <condition>
      <p>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.
      </p>
    </condition>
    <diagnostics>
      <diagnostics-general>
        <table frame="all" id="table_r2k_4sw_nlb">
          <tgroup cols="2">
            <colspec colname="c1" colnum="1" colwidth="1*"/>
            <colspec colname="c2" colnum="2" colwidth="1*"/>
            <thead>
              <row>
                <entry>Symptom</entry>
                <entry>Probable Cause</entry>
              </row>
            </thead>
            <tbody>
              <row>
                <entry>Clunking noise on bumps only</entry>
                <entry><xref href="#./checkstruts"/></entry>
              </row>
              <row>
                <entry>Continuous clunking noise</entry>
                <entry><xref href="#./checkballjoints"/></entry>
              </row>
            </tbody>
          </tgroup>
        </table>
      </diagnostics-general>
    </diagnostics>
  </troublebody>
</troubleshooting>
```

```

        </row>
        <row>
            <entry>Ticks when in neutral</entry>
            <entry><xref href="#./checkexhaust"/></entry>
        </row>
        <row>
            <entry>Ticks only in reverse</entry>
            <entry><xref href="#./checkbrakes"/></entry>
        </row>
        <row>
            <entry>Ticks in turns and curves</entry>
            <entry><xref href="#./checkcvjoint"/></entry>
        </row>
        <row>
            <entry>Only ticks when cold</entry>
            <entry><xref href="#./checkcatalyticconverter"/></entry>
        </row>
        <row>
            <entry>Ticks only at slow speed</entry>
            <entry><xref href="#./checkwheels"/></entry>
        </row>
    </tbody>
</tgroup>
</table>
</diagnostics-general>
</diagnostics>
<troubleSolution>
    <remedy id="checkstruts">
        <title>Checking your struts</title>
        <steps>
            <step>
                ...
            </step>
            ...
        </steps>
    </remedy>
</troubleSolution>
<troubleSolution>
    <remedy id="checkballjoints">
        <title>Checking your ball joints</title>
        <steps>
            <step>
                ...
            </step>
            ...
        </steps>
    </remedy>
</troubleSolution>
<troubleSolution>
    <remedy id="checkexhaust">
        <title>Checking your exhaust</title>
        <steps>
            <step>
                ...
            </step>
            ...
        </steps>
    </remedy>
</troubleSolution>
<troubleSolution>
    <remedy id="checkbrakes">
        <title>Checking your brakes</title>
        <steps>
            <step>
                ...
            </step>
            ...
        </steps>
    </remedy>
</troubleSolution>
<troubleSolution>
    <remedy id="checkcvjoint">
        <title>Checking your CV joints</title>
        <steps>

```

```

        <step>
            ...
        </step>
        ...
    </steps>
</remedy>
</troubleSolution>
<troubleSolution>
    <remedy id="checkcatalyticconverter">
        <title>Checking your catalytic converter</title>
        <steps>
            <step>
                ...
            </step>
            ...
        </steps>
    </remedy>
</troubleSolution>
<troubleSolution>
    <remedy id="checkwheels">
        <title>Checking your wheels</title>
        <steps>
            <step>
                ...
            </step>
            ...
        </steps>
    </remedy>
</troubleSolution>
</troubleshooting>

```

Figure 2: Adding step-by-step diagnostics to a troubleshooting topic

The following code sample shows how a troubleshooting topic would provide step-by-step instructions to help a user determine which cause and therefore which solution applies to them.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE troubleshooting PUBLIC "-//OASIS//DTD DITA 2.0 Troubleshooting//EN"
"troubleshooting.dtd">
<troubleshooting>
    <title>
    <shortdesc>
    <prolog/>
    <troublebody>
        <condition>
        <diagnostics>
            <diagnostics-steps>
                <steps>
                    <step>
                        <cmd>Open the command prompt and type <userinput>ipconfig</userinput>.</
cmd>
                        <info>
                            <p>The Default Gateway (listed last) is your router's IP. Your
computer's IP
address
address
address. See
                            <xref href="#./ipaddress"/></p>
                        </info>
                    </step>
                    <step>
                        <cmd>If your address does not start with 169, type <userinput>tracert
8.8.8.8</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 command
<cmdname>nslookup</cmdname> 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>
<troubleSolution>
    <remedy id="ipaddress">
        <title>Resetting Your IP Address</title>
    <steps>
        <step>
            ...
        </step>
        ...
    </steps>
    </remedy>
</troubleSolution>
<troubleSolution>
    <remedy id="resetnetwork">
        <title>Resetting Your Local Network</title>
    <steps>
        <step>
            ...
        </step>
        ...
    </steps>
    </remedy>
</troubleSolution>
</troubleshooting>

```