

An OASIS DITA Adoption Technical Committee Publication

DITA 1.3 Feature Article: Using DITA 1.3 Troubleshooting

Author: Bob Thomas
On behalf of the DITA Adoption Technical Committee

Date: 12 May 2014

This is a Non-Standards Track Work Product and is not subject to the patent provisions of the OASIS IPR Policy.

Table of Contents

- Using DITA 1.3 Troubleshooting..... 5
 - Troubleshooting information..... 5
 - DITA 1.3 troubleshooting features.....6
 - Troubleshooting-topic examples..... 6
 - Simple troubleshooting topic.....6
 - Multiple solutions troubleshooting topic.....10
 - Alarm troubleshooting topic..... 13
 - Embedded troubleshooting examples.....17
 - Task-related..... 17
 - Troubleshooting note-type..... 19
 - Annotated troubleshooting template..... 20

OASIS (Organization for the Advancement of Structured Information Standards) is a not-for-profit, international consortium that drives the development, convergence, and adoption of e-business standards. Members themselves set the OASIS technical agenda, using a lightweight, open process expressly designed to promote industry consensus and unite disparate efforts. The consortium produces open standards for Web services, security, e-business, and standardization efforts in the public sector and for application-specific markets. OASIS was founded in 1993. More information can be found on the OASIS website at <http://www.oasis-open.org>.

The OASIS DITA Adoption Technical Committee members collaborate to provide expertise and resources to educate the marketplace on the value of the DITA OASIS standard. By raising awareness of the benefits offered by DITA, the DITA Adoption Technical Committee expects the demand for, and availability of, DITA conforming products and services to increase, resulting in a greater choice of tools and platforms and an expanded DITA community of users, suppliers, and consultants.

DISCLAIMER: All examples presented in this article were produced using one or more tools chosen at the author's discretion and in no way reflect endorsement of the tools by the OASIS DITA Adoption Technical Committee.

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

Copyright © 2014 OASIS. All rights reserved.

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

Document History

Revision	Date	Author	Summary
First Draft	2 February 2014	Thomas	Draft of the Committee Note -- Feature Article
Draft	22 May 2014	Thomas	Draft for committee vote

Using DITA 1.3 Troubleshooting

This article briefly describes the troubleshooting information type plus the DITA 1.3 features that support it. Several examples follow that show how to use the new troubleshooting features.

Troubleshooting information

This topic describes the troubleshooting information type.

What is troubleshooting?

Troubleshooting is an information type that provides corrective actions for changing the state of a product or a system to a state that is more desirable.

Simple troubleshooting

In its simplest form, troubleshooting information provides corrective actions that follows this pattern:

1. A condition or symptom. Usually the condition or symptom is an undesirable state in a system, a product, or a service that a reader may wish to correct.
2. A cause for the condition or symptom
3. A remedy for the condition or symptom that restores the system, product, or service to its normal state.

Complex troubleshooting

In complex cases, there may be more than one possible cause for a condition or a symptom. When this happens, each cause can be presented along with its associated remedy. These cause-remedy pairs serve as successive fixes users can try to eliminate an undesirable condition.

Embedded troubleshooting

Embedded troubleshooting information appears within tasks or descriptions. It is brief. Often, the condition or cause is implied by the information that surrounds it. Remedies can usually be conveyed with a single sentence.



Tip:

Extensive amounts of troubleshooting information ought never be embedded. Use troubleshooting topics instead.

Other corrective action information

Other corrective action information that follows the troubleshooting information type pattern are:

Alarm clearing When something goes wrong, a system returns an alarm from a predefined set of alarms.

Error resolution When something goes wrong, a system returns an error code from a predefined set of error codes.

Event response When a significant event occurs, a system returns an event from a predefined set of events. Some of these events, while not errors, are nonetheless undesirable states that may warrant a response.

DITA 1.3 troubleshooting features

This topic describes DITA 1.3 support for the troubleshooting information type.

What are the troubleshooting features?

DITA 1.3 introduced the following troubleshooting features:



Troubleshooting topic	A new “troubleshooting” topic type that models troubleshooting information-type semantics.
tasktroubleshooting	This section-like element is a place for specifying the corrective action to take when a task fails.
steptroubleshooting	This step sub-element is a place for specifying the corrective action to take when a step fails.
Troubleshooting note-type	The “trouble” note-type is available alongside other note-types such as “caution”, “note”, or “tip”. The “trouble” note-type contains incidental corrective action that pertains to its surrounding content.

Why DITA 1.3 troubleshooting matters

With the new DITA troubleshooting markup, writers can create tightly focused information that helps readers resolve specific problems. Writers can now easily follow consistent patterns for troubleshooting information. This consistency expresses itself in predictable organization, titles, and icon graphics that lets readers quickly locate troubleshooting information. Companies save money when their users resolve problems without contacting technical support.

How do you use it?

Learn more about how to use these features by reading through the examples that follow. Additionally, an annotated template for the troubleshooting topic appears at the end of this article.

Troubleshooting-topic examples

Simple troubleshooting topic

This is a basic troubleshooting-topic application.

Scenario

Name	Tripped circuit breaker
Description	The system is plugged in, the power switch is on, but the system will not start. This problem is external to the system, and it is almost always due to a tripped circuit breaker. The system is a low-power consumer product that runs on household electricity.

Discussion

We will use a troubleshooting topic with a single troubleSolution element to document the corrective action for this problem. In the tag view, pay close attention to the XML comments embedded within the topic markup. These comments contain usage advice.

Output

Before looking at the tag view, look at the output first:

System will not turn on

Everything looks right, but the system still does not start.

Condition

The system is plugged in, the power switch is on, but the system will not start.

Cause

This problem is usually due to power not being supplied to the system through the electrical outlet. Often, a circuit breaker has been tripped so that no power is available at the outlet.

Remedy

Warning:

If you do not know how to reset circuit breakers, do not attempt to fix this problem. Instead, find somebody who is qualified to do this for you.

1. Turn the system power switch to **OFF**
2. Reset the breaker
3. Turn the system power switch to **ON**

The system turns on.

Tag view

Collectively, the following series of XML editor screen-shots form a complete troubleshooting topic. They have been split into several segments for formatting reasons.

troubleshooting id="TrippedBreaker"

#comment

The topic title describes the problem from a user's point of view. The title should refer to a problem's symptoms rather than to its causes.

#comment

title

System will not turn on

title

shortdesc

Short Description: Everything looks right, but the system still does not start.

shortdesc

prolog metadata **Metadata:**

keywords **Keywords:** #comment

Pay close attention to indexing. Often, this is how a user will find a topic.
Try to imagine all of the terms where a user might look.

#comment

indexterm **System will not turn on** indexterm

indexterm **Troubleshooting**

indexterm System will not turn on indexterm indexterm

indexterm **Help**

index-see *See: Troubleshooting* index-see indexterm

indexterm **Broken**

index-see *See: Troubleshooting* index-see indexterm

indexterm **Turn on**

indexterm System will not turn on indexterm indexterm keywords metadata

prolog

troublebody condition #comment

Use "Condition" for this title unless you have a good reason to do something else. Whatever you decide, you should be consistent from one troubleshooting topic to the next. This helps your user rapidly recognize troubleshooting information.

#comment

title **Condition** title

#comment

Condition content is a simple elaboration on what has already appeared in the topic title and in the shortdesc. Only include information that is directly related to the condition or the symptom that the topic resolves.

#comment

p The system is plugged in, the power switch is on, but the system will not start. p condition

#comment

Keep it simple with troubleSolution: one cause followed by one remedy. There may be cases where you have to do something different, but don't go there unless you absolutely must. #comment

troubleSolution cause #comment

Use "Cause" for this title unless you have a good reason to do something else. Consistency across topics is important.

#comment

title Cause title

#comment

Cause content should only describe the problem origins that are fixed by the cause's companion remedy element. #comment

p This problem is usually due to power not being supplied to the system through the electrical outlet. Often, a circuit breaker has been tripped so that no power is available at the outlet. p

cause

remedy #comment

Use "Remedy" for this title unless you have a good reason to do something else. Consistency across topics is important.

#comment

title Remedy title

#comment

The responsibleParty is the role of the person performing the remedy. This element is optional. Often, stylesheets will use this content as metadata and not output it directly. Do not use this element unless your stylesheets support it and you have a specific purpose in mind for using it.


#comment

responsibleParty electrician responsibleParty

#comment

The use of steps at this point in remedy is one of three mutually exclusive choices. Alternatively, you could use steps-unordered or steps-informal, but do that only if you have a good reason. Again, it important to maintain as much consistency as possible across troubleshooting topics.

#comment

steps stepsection note type="warning"  **Warning:**

p If you do not know how to reset circuit breakers, do not attempt to fix this problem. Instead, find somebody who is qualified to do this for you. p note stepsection

#comment

Often, identical steps appear elsewhere in tasks or in other troubleshooting topics. When this happens consider using conref. #comment

step **Step 1**

cmd Turn the system power switch to uicontrol **OFF** uicontrol cmd step

step **Step 2**

cmd Reset the breaker cmd step

step **Step 3**

cmd Turn the system power switch to uicontrol **ON** uicontrol cmd

stepresult **Step Result:**

p The system turns on. p stepresult step

steps remedy troubleSolution troublebody troubleshooting



Multiple solutions troubleshooting topic

This is a complex troubleshooting topic application.

Scenario

Name Cannot log in to the system.

Description A customer cannot log in to the system. The reasons for this could be: no account exists, the user forgot their user id, or the user forgot their password. The user needs to contact support if none of the remedies resolve this problem.

Discussion

We will use a troubleshooting topic with three troubleSolution elements to present a series of potential fixes (fallbacks) for this problem. The idea is that the user will try each troubleSolution until the problem is resolved. The first title in each troubleSolution is in the cause element, and it briefly describes the cause instead of using a consistent title such as “Cause.” In multiple troubleSolution scenarios, descriptive labeling helps users diagnose problems quicker. Therefore, it is more important than using a consistent title.

No condition element is used here. That is because all pertinent information about the condition has already been given in the topic title and in the shortdesc.

The first troubleSolution contains the “no account exists” cause/remedy. This troubleSolution must be first because there is no point in a user trying to reset account credentials for a non-existing account. The second troubleSolution contains the “forgotten user id/forgotten password” cause/remedy. The third troubleSolution contains the final fallback: calling customer support. In practical applications, this third troubleSolution would be reused through confref.

Output

Cannot log in to the system

The system rejects a user ID or user password.

No account exists

The system requires each user to have an account to access the system.

Remedy

1. Have your customer order number available.
2. Go to <https://nnn.nnn.nnn/IneedAnewAccount.jsp> to set up a new account.

Forgotten user ID or forgotten password

Forgotten user IDs and forgotten passwords can be reset over the internet.

Remedy

Go to <https://nnn.nnn.nnn/FixMyCredentials.jsp> to retrieve your user ID or to reset your password.

Contact support

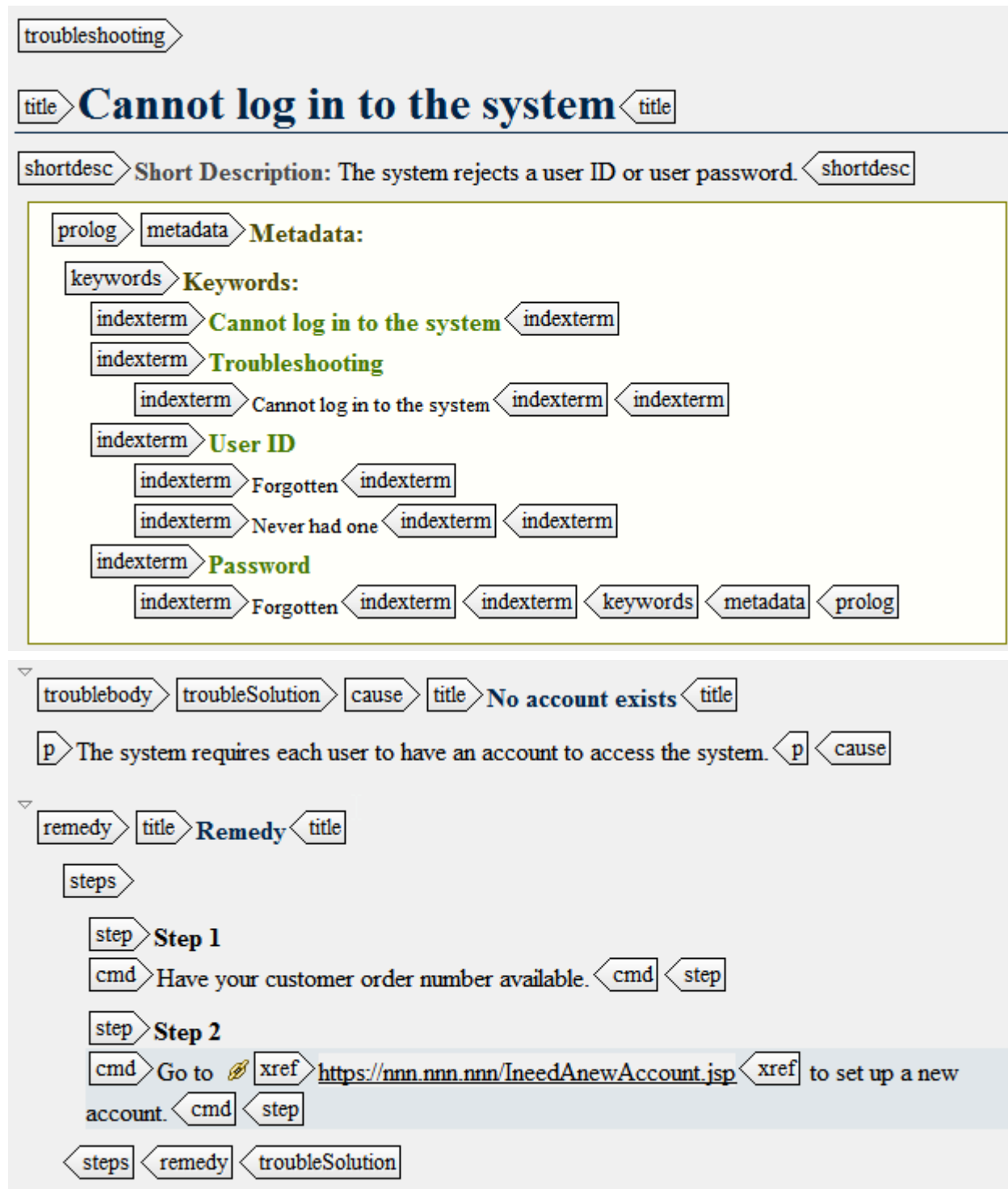
There must be some other cause for this problem.

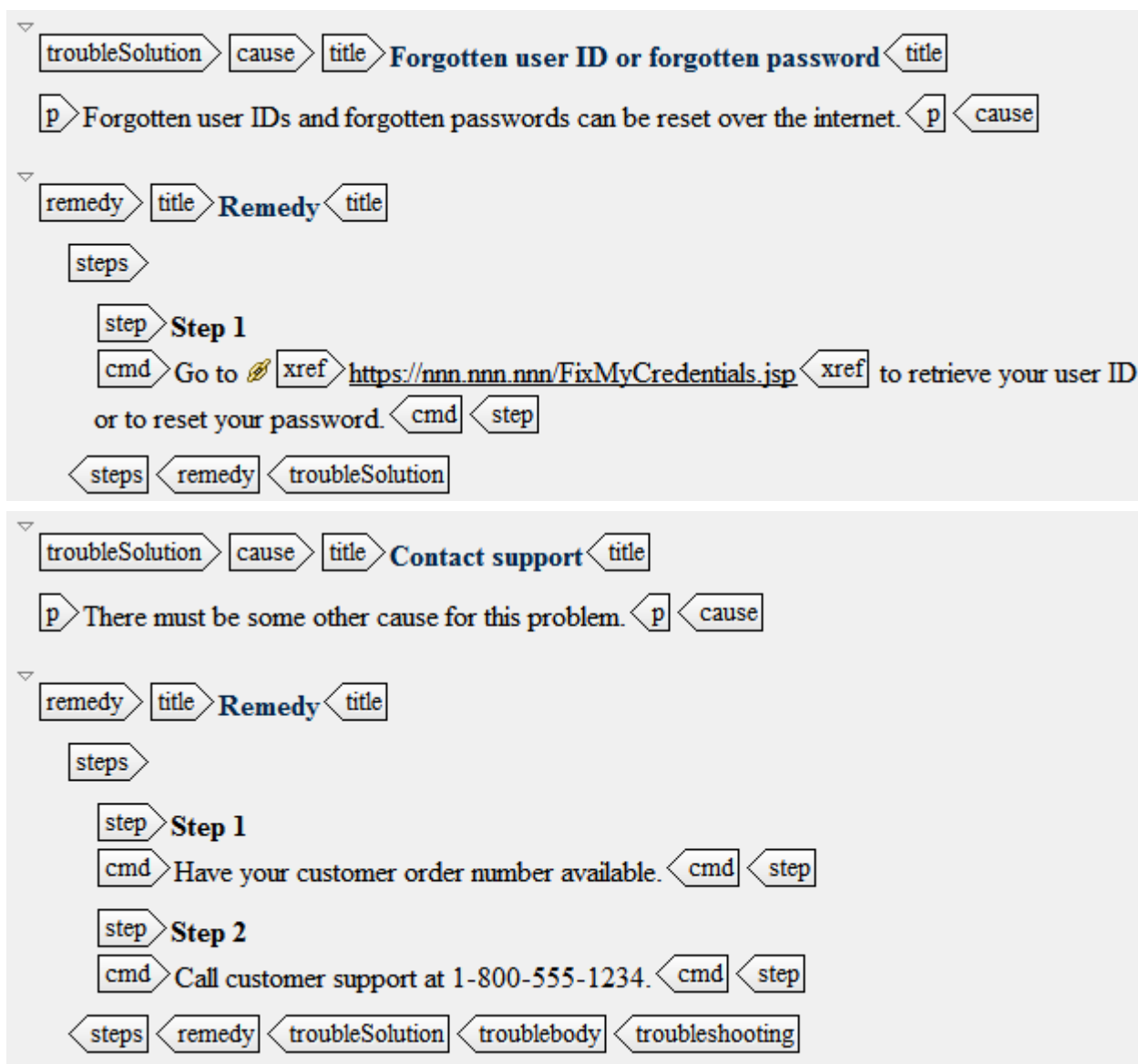
Remedy

1. Have your customer order number available.
2. Call customer support at 1-800-555-1234.



Tag view





Alarm troubleshooting topic

This shows how troubleshooting-topic can contain alarm information.

Scenario

Name Alarm S503

Description The system issues alarm S503 whenever its connection with the network controller has been unavailable for more than 2 minutes. This is an intermediate-level alarm. The impact to the system is moderate. Circuit card TN2037 provides the system's LAN connectivity.

Possible causes for this problem are: 1) TN2037 is in the off-line state, 2) the wiring between the TN2037 and the network controller may have disrupted, 3) the network controller is experiencing system problems, or 4) the TN2037 is defective.

Discussion

We will use a troubleshooting topic with four troubleSolution elements to document this alarm along with its associated corrective actions. Note the use of definition list within the condition element to organize the alarm's classification information.

Output

S503 - communication link failure

The communication link between the system and the network controller has been out of service for more than two minutes.

Condition

Alarm code	S503
Level	Intermediate
Impact	Moderate

The system issues alarm S503 whenever its connection with the network controller has been unavailable for more than 2 minutes. Circuit card TN2037 provides the system's LAN connectivity.

TN2037 is off-line

The TN2037 provides system connectivity to the LAN which is connected to the network controller. If the TN2037 is in the "off-line" state, no communication can happen between the system and the network controller.

Remedy

1. Log in to the system administration interface.
2. Go to **Tools > Administration > Network**.
3. Select **Restart the network interface**

Network controller is having problems

There may be problems at the network-controller end of the link.

Remedy

Contact the network-controller administrator to find out whether there are problems with the network controller.

LAN wiring is defective

The electrical connection between the system and the network controller may be defective.

Remedy

1. Perform a [Network link integrity test](#).
If the network link passes its test, quit these remedy steps and proceed to [Defective TN2037](#).
2. If the link is defective, replace the LAN cable between the TN2037 and the Ethernet hub.

Defective TN2037

Circuit card TN2037 is defective, and it must be replaced.

Remedy

Elided for this example. In real output this would be the first of several steps.

Tag view

troubleshooting id="AlarmTroubleshootingTopicSource"

title S503 - communication link failure title

shortdesc Short Description: The communication link between the system and the network controller has been out of service for more than two minutes. shortdesc

troublebody condition title Condition title

dl dentry dt Alarm code dt

dd p S503 p dd dentry

dentry dt Level dt

dd p Intermediate p dd dentry

dentry dt Impact dt

dd p Moderate p dd dentry dl

p The system issues alarm S503 whenever its connection with the network controller has been unavailable for more than 2 minutes. Circuit car TN2037 provides the system's LAN connectivity.

p condition

troubleSolution cause title **TN2037 is off-line** title

p The TN2037 provides system connectivity to the LAN which is connected to the network controller. If the TN2037 is in the "q off-line q" state, no communication can happen between the system and the network controller. p cause

remedy title **Remedy** title

responsibleParty administrator responsibleParty

steps

step **Step 1**

cmd Log in to the system administration interface. cmd step

step **Step 2**

cmd Go to menucascade uicontrol **Tools** uicontrol uicontrol **Administration**
uicontrol uicontrol **Network** uicontrol menucascade . cmd step

step **Step 3**

cmd Select uicontrol **Restart the network interface** uicontrol cmd step

steps remedy troubleSolution

troubleSolution cause title **Network controller is having problems** title

p There may be problems at the network-controller end of the link. p cause

remedy title **Remedy** title

responsibleParty administrator responsibleParty

steps

step **Step 1**

cmd Contact the network-controller administrator to find out whether there are problems with the network controller. cmd step

steps remedy troubleSolution




Embed troubleshooting examples

Task-related

This section contains examples showing streptroubleshooting and tasktroubleshooting

Discussion

The steptroubleshooting and tasktroubleshooting elements were designed to give users immediate, brief information about how to resolve problems when an undesired  result happens. This means writers should limit content to discussing condition, cause, and then perhaps a single-step. If lists containing instructions are present in the content, that is too much. The writer should re-design the content so that the instructions are in a separate task topic or are part of a troubleshooting topic. The writer should then refer the user to that other topic.

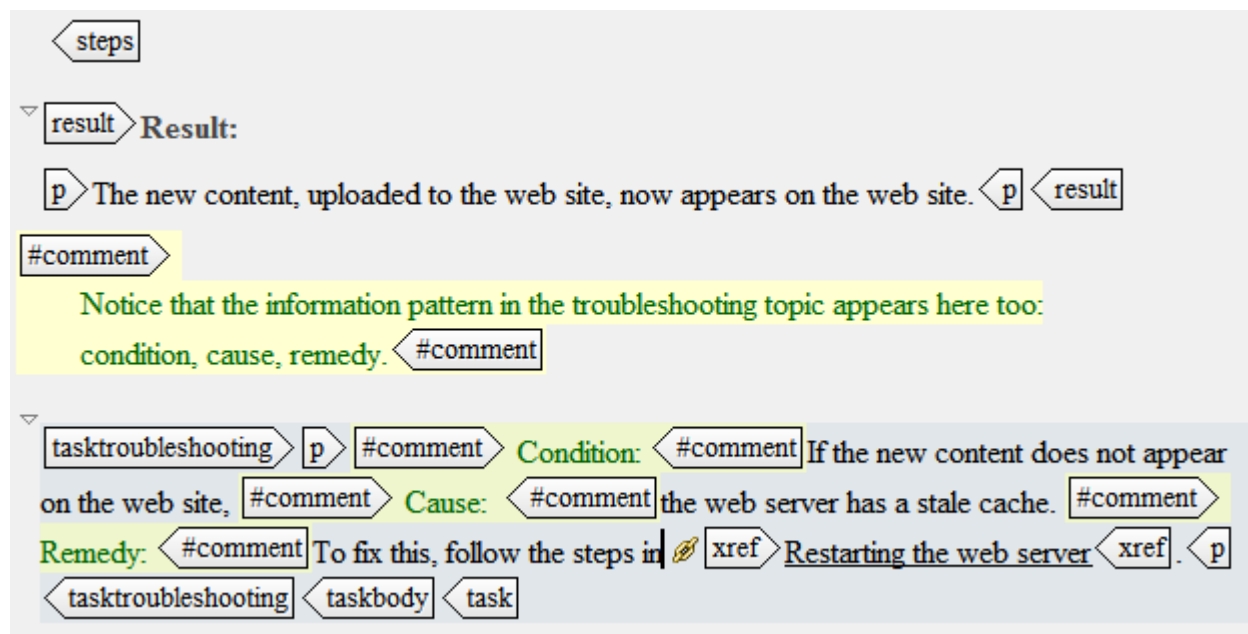
Task troubleshooting example

Scenario

Name Web site update fails

Description A customer follows the steps in a task for updating web site content, but the web site is not showing the new content. This could be due to a stale web server cache. The customer should restart the web server.

Tag view



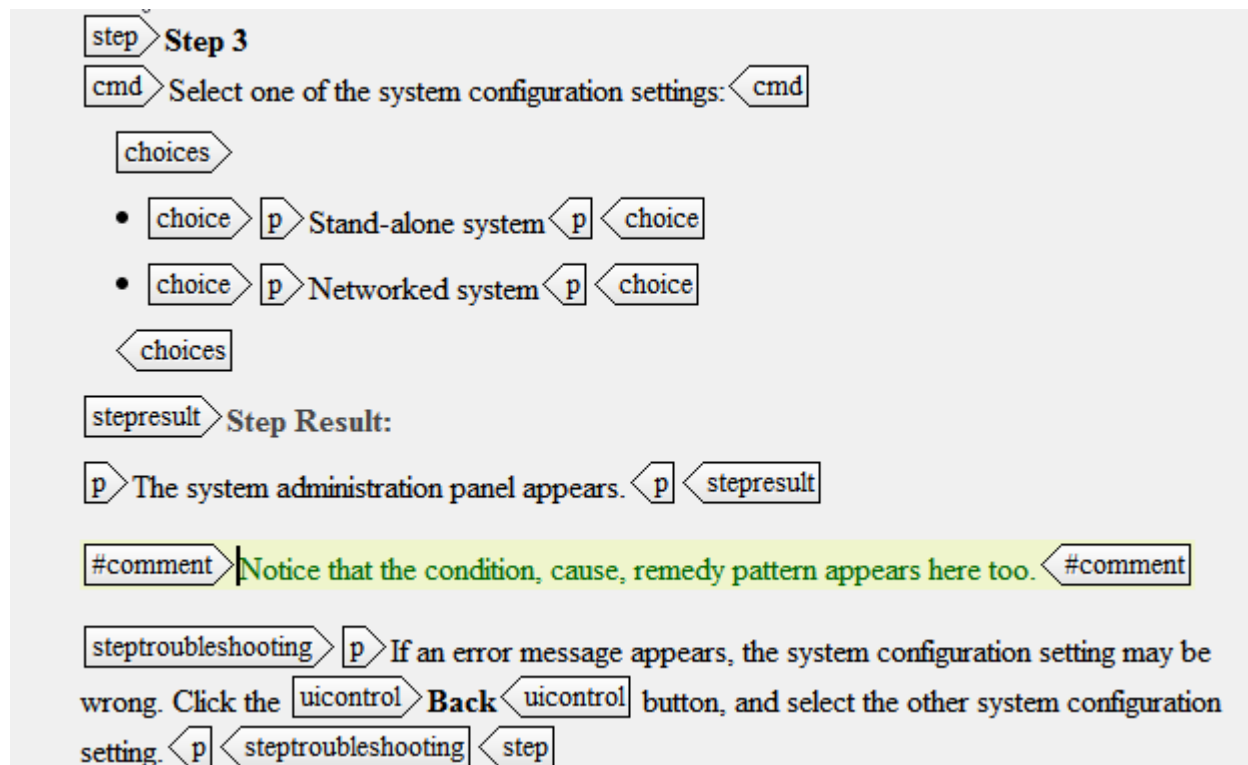
Step troubleshooting example

Scenario

Name Wrong option in step

Description A user has to select one of two system configuration settings in a step. If they select the wrong one, an error message displays. To recover, they need to click the **Back** button and select the other configuration setting.

Tag view



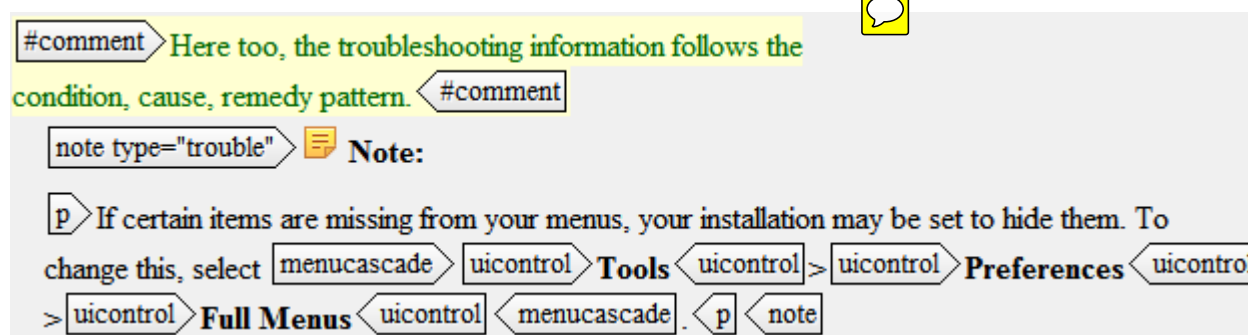
Troubleshooting note-type

Scenario

Name Cannot see all menu items

Description By default, a software product hides certain advanced menu items, however the product's documentation describes all of menus items. Customers following the documentation along on their systems have become confused when they did not see all of the menu items that were being described. The fix for this is to have the customer change their preferences to "Show Full Menus".

Tag view

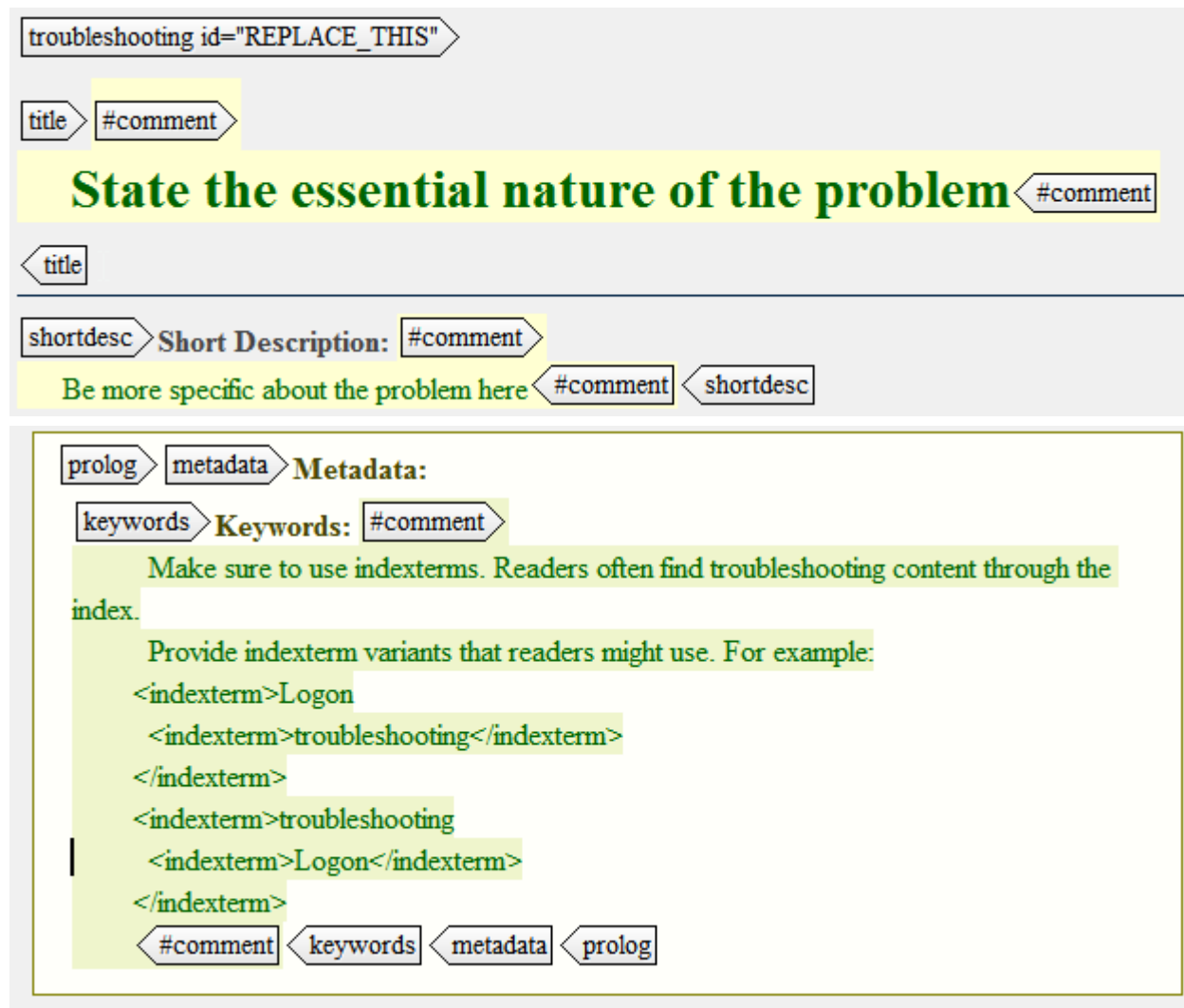


Annotated troubleshooting template

This is a DITA troubleshooting topic template with usage comments.

XML comments describe each element and how to use it. Formal descriptions for each troubleshooting element can be found in the *Darwin Information Typing Architecture (DITA) Version 1.3 OASIS Standard Language Specification* section.

Tag view





#comment

You can use **steps-unordered** or **steps-informal** instead of **steps**.

#comment

steps

#comment

Remember that you can use **conref** to bring in steps from tasks or other troubleshooting topics.

#comment

step **Step 1**

cmd

cmd

cmd

step

steps

remedy

troubleSolution

troublebody

troubleshooting

XML view

Here is the template's XML. It can be copied and pasted into an XML editor.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE troubleshooting
  PUBLIC "-//OASIS//DTD DITA Troubleshooting//EN"
  "troubleshooting.dtd">
<troubleshooting id="REPLACE_THIS">
  <title><!-- State the essential nature of the problem --></title>
  <shortdesc><!-- Be more specific about the problem here --></shortdesc>
  <prolog>
    <metadata>
      <keywords><!--
        Make sure to use indexterms. Readers often find troubleshooting content through the index.
        Provide indexterm variants that readers might use. For example:
        <indexterm>Logon
        <indexterm>troubleshooting</indexterm>
      </indexterm>
      <indexterm>troubleshooting
      <indexterm>Logon</indexterm>
      </indexterm>
      -->
    </keywords>
  </metadata>
</prolog>
  <troublebody>
    <condition><!--
      The topic title and the shortdesc should have already told your reader a lot about the condition
      that this topic seeks to fix. Use condition to expand upon that. Do not simply repeat what you have
      already put into the title and the shortdesc. Condition is the appropriate place to put impact
      and severity information. If multiple solutions exist and their relationships with each other are
      complex, you can discuss that here.-->
      <title><!-- Optional title. Use "Condition". --></title>
      <p></p>
    </condition>
    <troubleSolution><!--
      troubleSolution is meant to hold pairs of cause and remedy. Occassionally, you might have cause
      without remedy or remedy without cause, but that should be rare. Be sure to order mutiple
      troubleSolution elements in a sequence that makes sense. For example, order them by the liklihood
      of a cause occuring. You may wish to deviate from that if a remedy for a less likely cause is much
      easier to try. Remember to use conref for troubleSolution elements that are the same across
      multiple troubleshooting topics.-->
      <cause>
        <title><!--
          Optional title. Use "Cause". For topics with mutiple troubleSolutions, state the essential
          nature of this particular cause instead of just using "Cause".-->
        </title>
        <p></p>
      </cause>
```

```

<remedy>
  <title><!-- Optional title. Use "Remedy" or "Solution". -->
  </title>
  <responsibleParty><!--
    Optional. Use this element to indicate the role of who ought to be performing the steps in the
    remedy. Here are some examples: "engineer", "customer", "field-support".
  -->
  </responsibleParty>
  <!-- You can use steps-unordered or steps-informal instead of steps. -->
  <steps>
    <!-- Remember that you can use conref to bring in steps from tasks or other troubleshooting topics. -->
    <step>
      <cmd/>
    </step>
  </steps>
</remedy>
</troubleSolution>
</troublebody>
</troubleshooting>

```