

Introduction:

Think about the last time you used the documentation for a product. What were you looking for? Most likely, troubleshooting information.

David Farkas said at WinHelp 97: 30% of people use help for troubleshooting information.

Next question: How easy was it to find the information? People typically look in the documentation for no more than a minute before giving up and calling technical support.

This is why I think that out of all all the changes Microsoft made in Windows 95 help, the troubleshooter format is probably the most useful introduction they made, from the user's standpoint.

Purpose of class:

Teach you how to create troubleshooters.

The actual designing and compiling of a troubleshooter is easy as far as help files go. The real challenge is to decide what information to put in it and how to organize it.

Therefore, this program is about *content* rather than technology.

To some, the content issues may seem obvious, but you'd be amazed how many get it wrong. Also, mastering the content issues will make it easier for you to develop the help file.

Overview

- What troubleshooters do
- Steps for selecting and organizing information
- Writing text for a troubleshooter
- Creating troubleshooters in different help formats
- Making troubleshooters easy to access
- Maintaining troubleshooters

What troubleshooters do

- Guide users to a solution through a series of questions about the problems they have
- Provide a variety of paths for different types of problems
- Contain procedures for resolving problems
- Contain a limited set of information

Guide users: Like an interview between the user and support person: “What type of problem do you have?” “What happens when you do this?”

Variety of paths: A good troubleshooter anticipates different types of problems and give users a variety of options. It should also include feedback that provides additional options if what the user tries doesn’t work.

Procedures: It tells users how to fix the problem. In Windows 95 or HTML-based Help, it can also have links to applications they can use to fix the problem. (Perhaps, one day, it will fix it for them.) Not only does this help users, it educates them.

Limited set of information: Because of all the links you need to create, it doesn’t make sense to put a huge document database in a troubleshooter. You need to select relevant information customers can use in the field.

Examples of troubleshooters

Real-life example: Microsoft Direct Cable Connection troubleshooter:

Start it from the Windows help system.

Model of a troubleshooter: simple questions that guide you to the solution.

Shows the procedure.

Gives user opportunity to provide feedback: "Did this fix the problem?"

Doesn't take up much screen space.

Platinum SQL Installation Troubleshooter:

Stand-alone troubleshooter.

Start it from the application's menu.

Buttons for copy and print.

Steps for selecting and organizing information

1. Choose which problems you want to document in the troubleshooter.
2. Group related problems.
3. Determine symptoms, causes, and solutions for each problem.
4. Map paths to guide users to the solution.

Step 1: Choose problems to include in the troubleshooter

- Problems that can be fixed while the computer is on
- Problems users can fix themselves
- Problems that have proven solutions

Problems that can be fixed while the computer is on: Of course, if you can't boot the computer, you can't access the troubleshooter. Troubleshooting problems that prevent the computer or operating system from starting should be documented in the printed manual.

Problems users can fix themselves: Don't document any troubleshooting procedures that can cause users to damage their data or themselves. Refer to technical support or authorized service centers. Also, avoid problems that are difficult for your average user to correct. More complex solutions should be handled by support or an authorized service center.

Problem that have proven solutions: Every so often, a support person or programmer has some hack that fixes a problem. Don't document it unless it has been tested. Nothing is more frustrating than giving a solution that doesn't work (or causes more damage than before).

Step 2: Group related problems

- Remember the 7 ± 2 rule
- Organize according to how user encounters problems

Why group?:

Makes information easier to find. Users can go to the information they need instead of sifting through a large list of problems.

Makes the troubleshooter less intimidating. User only needs to go through a few options instead of a lot.

Rules for grouping:

7 ± 2 rule: The limit of items most people can retain in short-term memory. (Think of Seven Dwarfs, nine in the Brady Bunch, etc.)

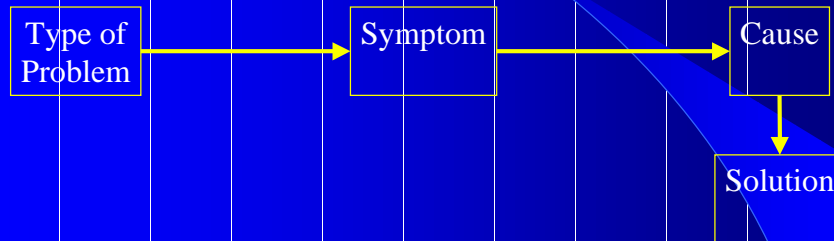
Organize according to how user encounters problems: Some problems may have the same technical cause, but the user encounters them in completely different ways. For example, the user may have problems with fonts according to the printer driver setup. However, you would organize this as a font problem rather than a printer driver problem. The user won't know it's a printer driver problem until you show him or her how to correct it.

Step 3: Determine symptoms, causes, and solutions

- Each problem consists of a symptom, the possible causes of the symptom, and a solution for fixing it

Symptom	Cause	Solution
The document does not print.	The printer is off line.	Press the Online button on the printer and try printing again.

Step 4: Map paths to guide users to the solution

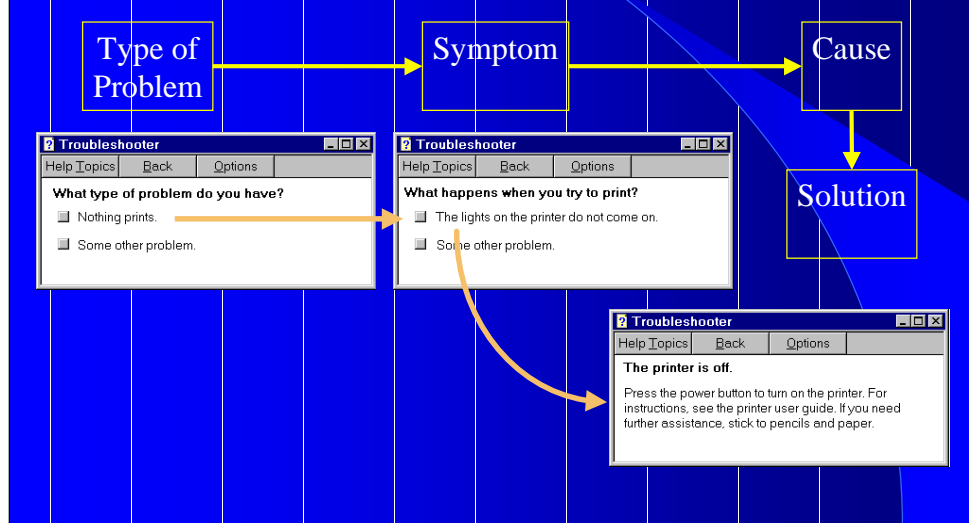


Troubleshooting is a four-step process:

1. User identifies the type of problem they are having.
2. User identifies the symptoms they are experiencing.
3. User learns about the cause of the problem.
4. User learns the solution of the problem.

These steps correspond...

Step 4: Map paths... (Continued)



...to screens in a troubleshooter:

The first screen enables users to select the type of problem they have.

The second screen enables users to select the symptoms that match their problem.

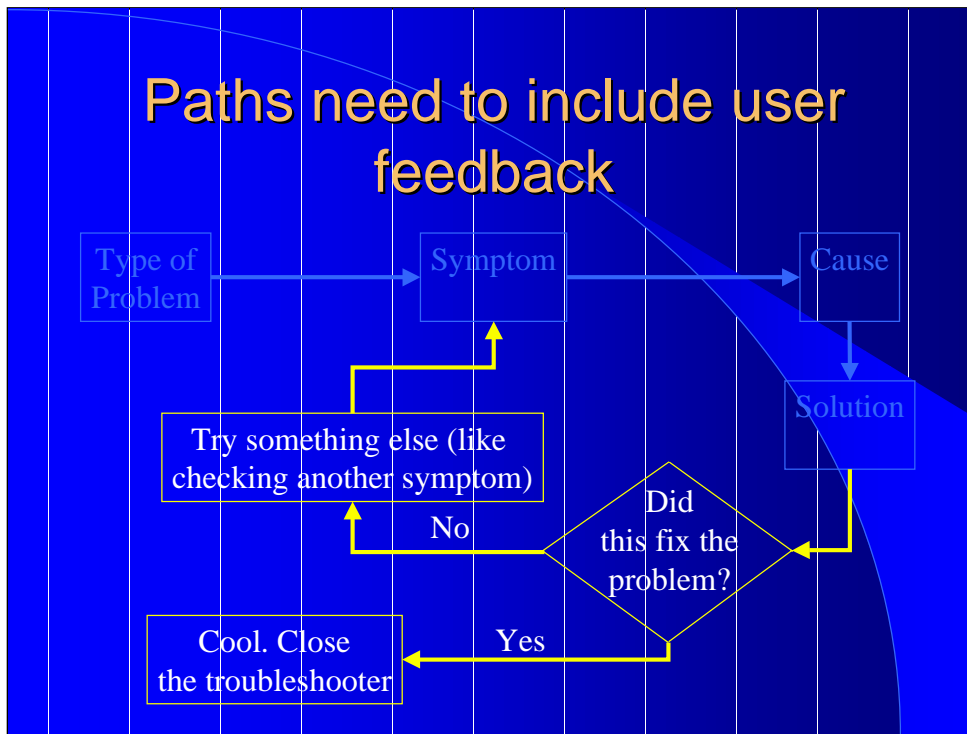
The third screen usually combines an explanation of the cause and the solution.

Complex problems might require additional screens but limit to 7 ± 2 – preferably less. Need to avoid users getting “lost in hyperspace” by giving them too many screens until they reach the solution

Donna Timpore: “No more than three clicks to find the answer to any question.”

Consider drawing flowcharts for your troubleshooters. These might help you make changes later.

Paths need to include user feedback



Feedback: Troubleshooter needs to add a way for users to indicate if the solution fixed the problem:

If so, close the troubleshooter.

If not, go to a screen that suggests other symptoms.

Example: Direct Connect Cable Troubleshooter.

Questions/Exercise

Questions?

Run audience through an example.

Ask audience member to identify a problem

Create a troubleshooter in ForeHelp based on the troubleshooting information he or she gives.

Writing text for a troubleshooter

- Keep it short and simple
- Focus on the task
- Keep questions simple

Keep it short and simple: Remember your readers are short on time and patience when they are using a troubleshooter. Write text so it can be read easily.

Focus on the task: Troubleshooters are task-oriented. Provide only the information the user needs to solve the problem. Avoid adding descriptions or background information unless it is necessary to help the user fix the problem.

Keep questions simple: Provide questions that can be answered with a simple yes or no or a selection from a multiple choice. No essay questions. If you need to ask a user several questions, put each one on a separate screen. This makes it easier for users to answer the questions and for you to develop the help. (Imagine how hard it would be to process different answers at the same time.) You can also skip questions based on the user's previous responses.

Creating troubleshooters in different help formats

- General rules
- Tips for WinHelp
- Tips for HTML (all forms)

General rules for troubleshooters

- Use as little of the screen as possible
- Keep the screen design clean and uncluttered
- Make copy and print commands readily available

Tips for WinHelp troubleshooters

- Use “task topic window” format (could be primary or secondary window)
- Consider adding shortcuts to applications or applets
- Consider putting Print and Copy buttons on toolbar

Task topic window format:

Normally, you would use a secondary help window. [Ask if audience understands what that means.]

But, if you are doing a stand-alone troubleshooter, it's better to put it in a primary window. You can set the colors and size to look like a secondary window.

Uses a little of the screen as possible. If you use Windows 95 help, use the Auto height option.

Uses the system color for tooltips -- typically light yellow (red 255, green 255, blue 226).

Shortcuts to applications and applets (Windows 95):

If there is a utility a user can use to fix a problem, add a shortcut to it.

Example: A shortcut to Printer Setup in the Control Panel.

Print and Copy buttons:

Not in Microsoft's standards, but a helpful addition to the window.

[Mention: You can download templates from my web site or the post-Conference web site.]

HTML troubleshooters

- Creative Labs

<http://www.creativehelp.com/>

- Quantex

<http://kb.quantex.com/tblshoot/trblmain.htm>

Preface:

HTML a good alternative – put troubleshooting information on your web site or use HTML-based help.

Examples from the Internet:

Creative Labs CreativeHelp: Good example of a troubleshooter as a knowledge base.

Quantex troubleshooter: Very plain, but easy to follow

Tips for HTML troubleshooters

- For HTML-based help, make the window as small as possible
- Use direct links, instead of having users select an option and then click Submit
- Keep design simple

Linking troubleshooters to web sites

- Provide updated information, patches, and extensive knowledge base
- Consider which information to put on the Web and which to keep on user's local drive
- Works with WinHelp and HTML-based help files

You can link a troubleshooter on the user's local drive to your web site.

Show an example.

How to use:

Use web for updated information, patches, knowledge base, and bug-report system.

Keep crucial information on the troubleshooter on the users' local drive in case they cannot reach your web site.

You can create links to the web site in either WinHelp (WinHelp 4.0 works best) or HTML-based help. You should provide some message in case the user is unable to link to your web site. For example, in the WinHelp 4.0 ExecFile macro, you can open a pop-up topic if the ExecFile macro fails. So, you can create an "unable to open web site" topic with a URL that the user can try later. (Show example from new Installation Troubleshooter.)

Making troubleshooters easy to access

- Put them into application menus
- Link to error message dialog boxes
- Dock in application windows (HTML Help)
- Use Active Desktop (IE 4.0/Windows 98)

Menu options:

Help menu

Right-click popup menus (William Meisheid's suggestion)

Start menu/Program Manager

Active Desktop:

Use WinBook's example.

It is an HTML page that is displayed as a background.

You can use it to:

Launch help files.

Provide a Quick Reference Card

Announce patches and updates in real time (if you can assume a constant Internet connection).

Maintaining troubleshooters

- Organize information so that it is easy to update
- Consider using flowcharts to identify areas to change

Conclusion

- Select and organize information properly
- Keep troubleshooters short and simple
- Make troubleshooters easy for users to access

Getting materials from today's class

How to get templates and slides:

- From my web site:

<http://members.aol.com/mastrn/>

Click on the Events button

- From the post-conference web site

Close with quote from Mary Deaton, KNOWware:

“Developing help has very little to do with the technologies we use and everything to do with understanding information structure, navigation architectures, usability, interface design, and instructional writing.”

Questions