Chapter 7: To Err Is Human: An Introduction to Debugging

Fluency with Information Technology
Third Edition

by
Lawrence Snyder

Precision: The High Standards of IT

- Precision in Everyday Life
  - Many people say "oh" rather than zero when giving a phone number
  - The listener makes the mental conversion because he knows phone numbers are numeric
  - Computer does not know that unless it has been programmed to know it

Exactly How Accurate Is "Precise?"

- Modem or database software can be programmed to make "oh" to zero corrections automatically because all digits will always be numbers
- In e-mail programs, both letters and numbers are allowed, so the computer can't be programmed to auto-correct. Users have to be careful

Lexical Structures

- *Field Inputs*: Information entered into boxes/forms on screen
  - Governed by *lexical structures* (rules about the legal form for input fields)
    - May limit symbols that can appear in specific positions, length of entry, etc.
    - May also be *loose*, allowing any sequence of symbols of any length

Debugging: What's the Problem?

- *Debugging*: The method of figuring out why a process or system doesn't work properly
  - Usually learned from experience
- Debugging in Everyday Life: People troubleshoot problems such as why their cars don't start
  - Usually involve working system with broken or worn-out part
- Debugging in Information Technology:
  - Working system might have wrong data or wrong configuration information or
  - There might be a logical design error
  - We will always begin by assuming a correct, working system

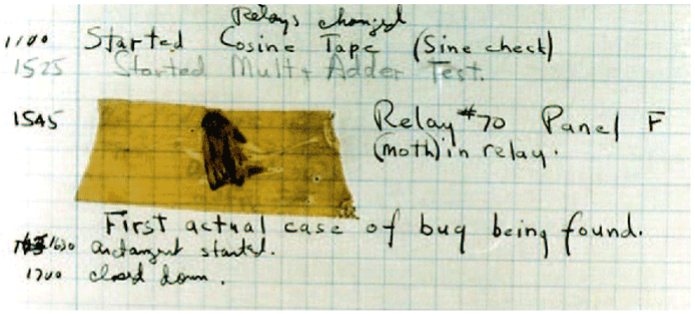


Figure 7.1 The Harvard Mark II logbook noting "First actual case of bug being found."

Whose Problem Is It?

- When we debug an information system, we are always part of the problem
  - We give the commands and the input
    - Operator or pilot error
  - Only other possible cause is broken system
- People don't knowingly make errors
  - We may think we did everything right because we misunderstand the system

Using the Computer to Debug

- A computer can't debug itself
- We can't debug it directly, either
- Error is internal to the computer
  - To get information about the error, we have to ask the computer what data it stored, etc.
- With faulty software we cannot fix, try to bypass error with alternative approach (*workaround*)

A Dialog About Debugging

- Debugging is solving a mystery
  - Whatdunit vs. whodunit
- Ask purposeful questions like:
  - Do I need more clues?
  - Are my clues reliable?
  - What is a theory to explain the problem?
- Better than aimlessly "trying stuff"

Debugging (cont'd)

- Steps to follow:
  1. Check that the error is reproducible
  2. Make sure you know what the problem is
     - If no mailing labels come out of printer, problem may be with printer or with program sending labels to printer or file containing addresses
  3. Check all the "obvious" sources of error
  4. Isolate the problem
     - Divide operation into those parts that are working and those that are not working
     - Think objectively—ask yourself if you've made a wrong assumption, etc.

Debugging Recap

- Make sure you can reproduce the error
- Determine exactly what the problem is
- Eliminate "obvious" causes (Is it plugged in?)
- Divide the process, separating out the parts that work from the part that doesn't (isolate the problem)
- When you reach a dead end, reassess your information; then step through the process again
- As you work through, make predictions about what should happen and verify that they are fulfilled

Butterflies And Bugs: A Case Study

- Imagine we've developed a simple HTML page
- Following is our goal page:

Our Goal Page:

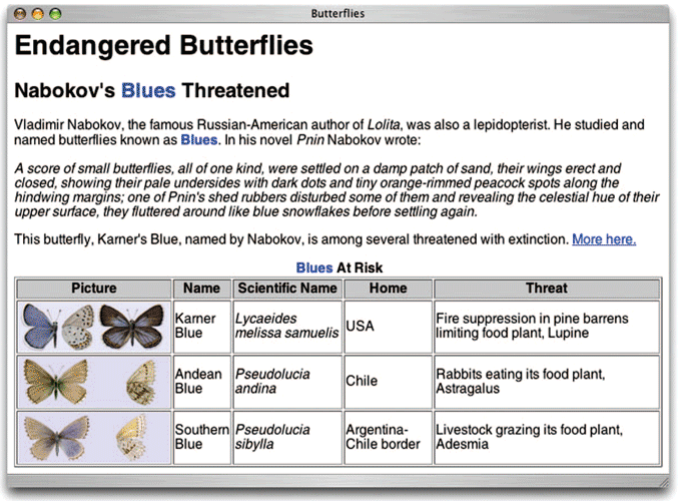


Figure 7.2 The intended Web page.

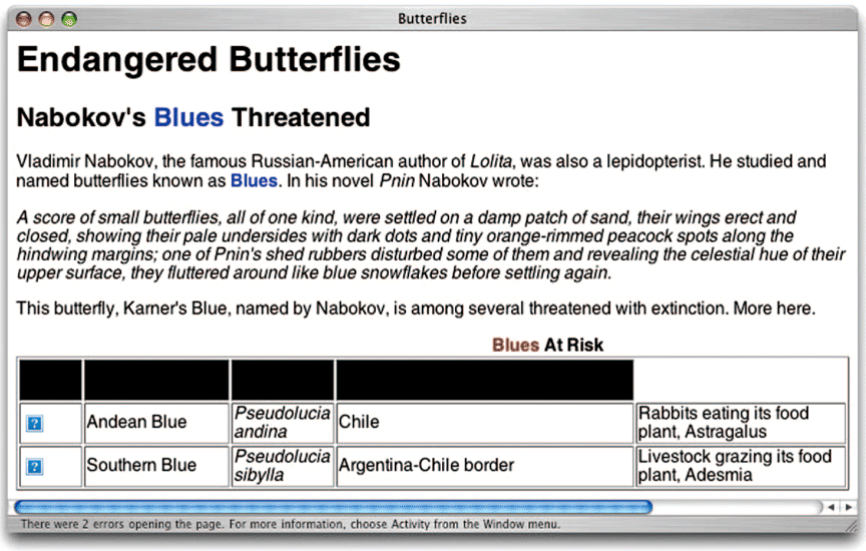


Figure 7.3 The Butterflies page displayed by the Safari browser.

The Actual Display in IE:

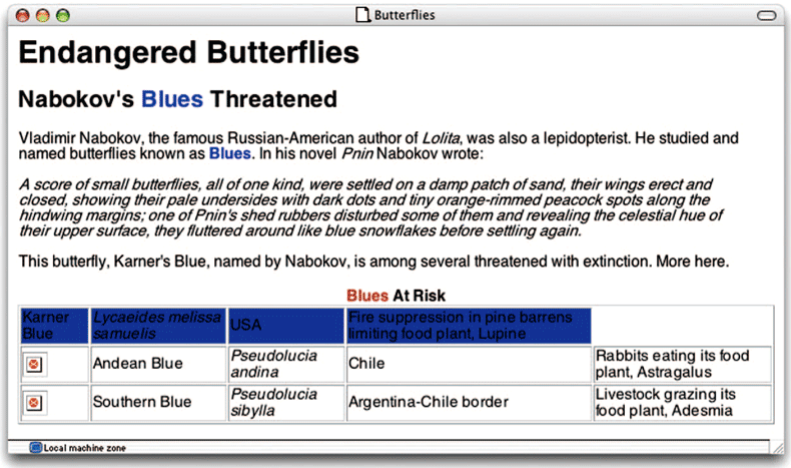


Figure 7.4 The Butterflies page displayed by the Internet Explorer browser.

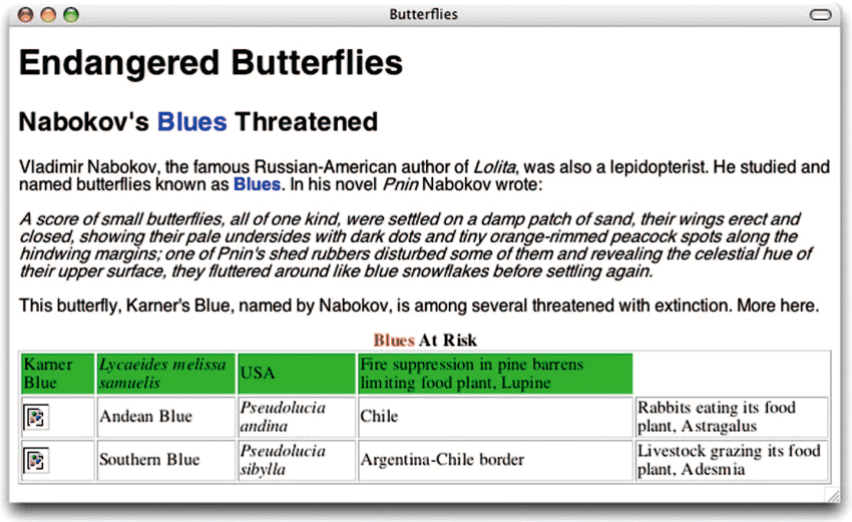


Figure 7.5 The Butterflies page displayed by the Firefox browser.

Butterflies and Bugs (cont'd)

- Displaying the page in Safari and Mozilla also produces errors
  - Test in multiple browsers!
- We can study the HTML very closely and "brain out" where the error is
- We will use the debugging strategy

Butterflies and Bugs (cont'd)

- First, be sure we can re-create the error
  - Reload the page
  - In this case, the results are the same
- Next, determine the problem exactly
  - Look at the displayed page and determine where the errors occur
  - Since there are multiple errors on different parts of the page, assume they are caused by different mistakes in the HTML
    - Try to fix first/top error on page first; save and refresh browser

The HTML for Our Page

```
<html>
   <head>
      <title>Butterflies</title>
   </head>
   <body><font face="Helvetica">
      <h1> Endangered Butterflies</h1>
      <h2>Nabokov's <font color="blue">Blues</font> Threatened</h2>
      <p>Vladimir Nabokov, the famous Russian-American author of
      <i>Lolita</i>, was also a lepidopterist. He studied and named butter-
      flies known as <b><font color=blue>Blues </font></b>. In his novel
      <i>Pnin</i> Nabokov wrote:</p>
      <p><i>A score of small butterflies, all of one kind, were settled on a
      damp patch of sand, their wings erect and closed, showing their pale
      undersides with dark dots and tiny orange-rimmed peacock spots along the
      hindwing margins; one of Pnin's shed rubbers disturbed some of them and
      revealing the celestial hue of their upper surface, they fluttered
      around like blue snowflakes before settling again. </i></p>
      <p>This butterfly, Karner's Blue, named by Nabokov, is among several
      threatened with extinction.
      <a href="http://www.libraries.psu.edu/nabokov/endan2.htm"> More
      here.</a></p>
      <table border>
         <caption><b><font color=blue">Blues</font> At Risk </b> </caption>
         <tr bgcolor="silver"/>
            <th>Picture</th>
            <th>Name</th>
            <th>Scientific Name</td>
            <th>Home</th>
            <th>Threat</th></tr>
         <tr><td><img src="lmeli.jpg"/></td>
            <td> Karner Blue</td>
            <td><i>Lycaeides melissa samuelis</i></td>
            <td>USA</td>
            <td>Fire suppression in pine barrens limiting food plant,
                Lupine</td></tr>
         <tr><td><img src="pandi.jpg"/></td>
            <td> Andean Blue</td>
            <td><i>Pseudolucia andina</i></td>
            <td>Chile</td>
            <td>Rabbits eating its food plant, Astragalus</td></tr>
         <tr><td><img src="psiby.jpg"/></td>
            <td> Southern Blue</td>
            <td><i>Pseudolucia sibylla</i></td>
            <td>Argentina-Chile border</td>
            <td>Livestock grazing its food plant, Adesmia</td></tr>
      </table>
      </font>
   </body>
</html>
```

Figure 7.6 Faulty HTML text for the Butterflies page.

Butterflies and Bugs (cont'd)

- Next, eliminate the obvious
  - In HTML, most common error is forgetting to close a tag
    - For example: with no closing tage
  - Make sure all quotes match (open and close)
    - In this case, the word **Blues** appears in red on the page. Why?

      <caption>Blues
    - Missing opening quotation mark around the attribute value "blue"
  - Why is the "More here" link not highlighted?
    - Space missing between <a and href

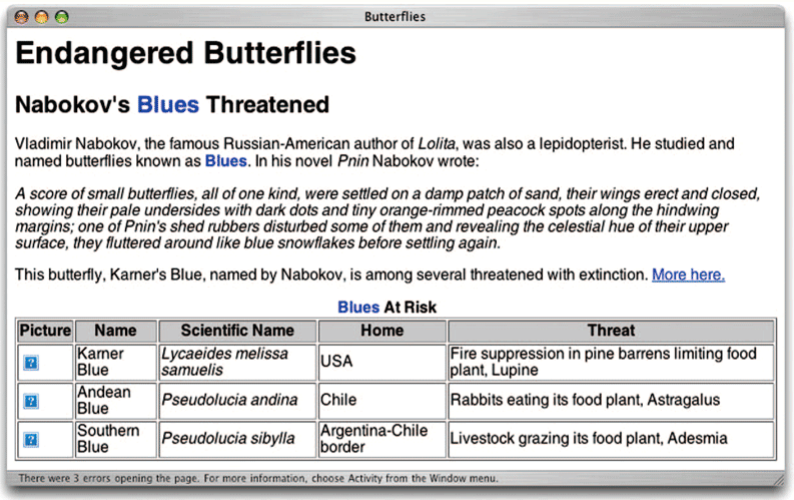


Figure 7.9 The Butterflies page after correcting the heading color.

Butterflies and Bugs (cont'd)

- Divide up the process
  - Separate the parts of the system that work from the parts that do not
    - First row of the table
    - Browsers and debugging systems for HTML can give us a color-coded version of the HTML, showing how it's being interpreted (for example, Notepad++)
      - View (page) source
    - We see the entire heading line is colored blue as if it's an attribute
    - The problem turns out to be a fancy quote that the browser cannot interpret

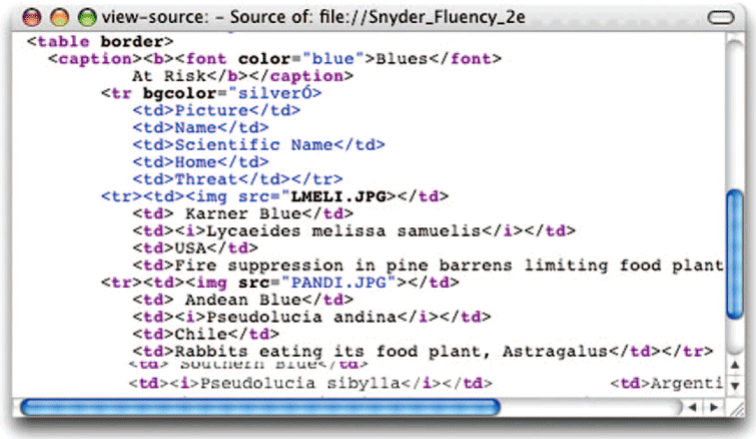


Figure 7.8 The Source View displayed by the Firefox browser for the start of the table.

Butterflies and Bugs (cont'd)

- Reassess
  - The pictures are still not displaying
  - We find that the image tags are well structured and the browser is interpreting them correctly
  - What is the problem?
    - To check that the images are ok, display in another document
    - To check that the JPEG specification is ok, put another image file in the document
    - Check the exact file name for the image, including capitalization
      - In this case, we find the filename extension is .jpeg, not .jpg as we have been using

Butterflies and Bugs (cont'd)

- Unnecessary changes:
  - During the debugging process, we made some unnecessary changes due to wrong conjectures about the error
  - Making unnecessary changes is typical in debugging
    - Sometimes we even make the situation worse by introducing new errors

Butterflies and Bugs (cont'd)

- Hiding other errors:
  - At first we thought we had three errors—bad caption, missing link, busted table
  - Because there were two things wrong with the table (messed up heading line and wrong file names specified) there were actually four errors
  - Because it is common for one error to hide another, always suspect there is more than one error

Butterflies and Bugs (cont'd)

- Viewing the source:
  - Most effective technique in our debugging was the View Source feature
  - In general, one of the most powerful debugging techniques is to find ways for the computer to tell us the meaning of the information it stores or the commands it executes
- Little errors, big problems:
  - The errors in the HTML code were tiny, but they had serious effects
  - We must be extremely precise

No Printer Output: A Classic Scenario

- Most systems we use, we don't create
- The software is very complicated. How do we troubleshoot a system we don't understand?
- Generally, software has been extensively tested before we come in contact with it
  - Standard operations are likely to be bug-free
- To illustrate debugging a system without understanding it, consider a common problem: You try to print a document and nothing comes out of the printer

Applying the Debugging Strategy

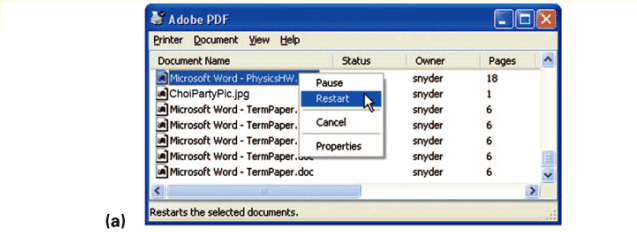
- Apply the debugging strategy:
  - Reproduce the error, understand the problem, check the obvious causes
  - Check the printer's control panel, the paper, the cartridges, cable connection, file to be printed, installation of printer driver, whether others can print (if it's a shared printer) and whether you can print a different document
    - Simple, quick checks
  - If this does not solve the problem, press on

Pressing On

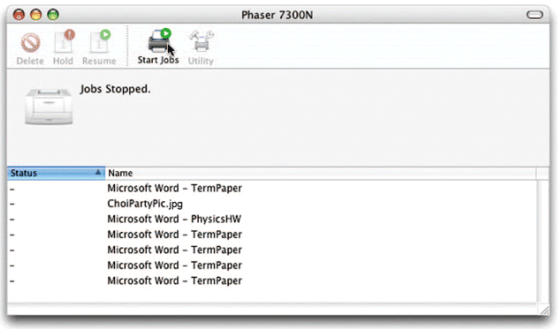
- Try to isolate the problem:
  - Configuration is correct because you've printed before
  - Try printing a simple document
    - Same problem
  - Think through the probable process
    - If the computer couldn't send the data to the printer, wouldn't it give a message to plug in the printer?
    - Look around for the stranded files
      - In the printing monitor's files, you find a list of files you've tried to print recently
      - Start > Printers and Faxes

The Print Queue

- You've found the print queue
- Computer's settings may tell it to "queue" converted files rather than printing them immediately
  - Pause Printing
- You may never know how this occurred, but you can still fix it by re-configuring the driver
  - Unclick Pause Printing or click Resume Printing



(a)



(b)

Figure 7.11 Printer queues for (a) Windows and (b) Mac operating systems.

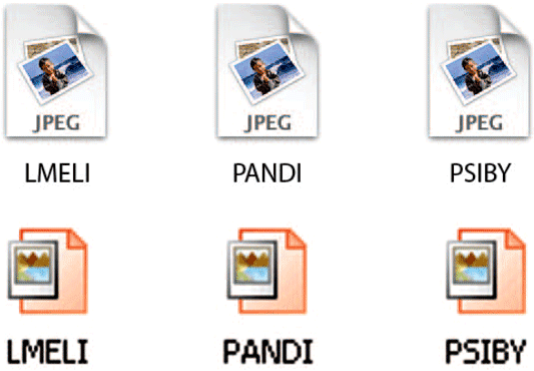


Figure 7.7 The image files for (a) Mac OS X and (b) Windows.

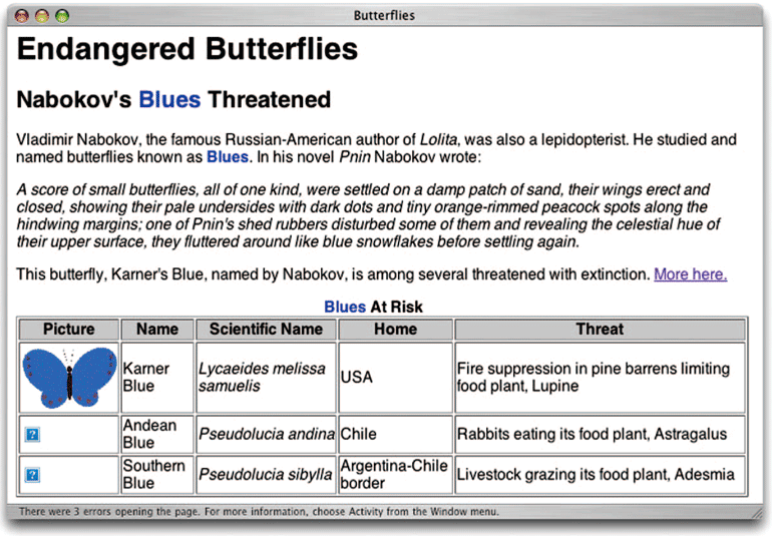


Figure 7.10 The table from the Butterflies page after revising with a new file.