Crash & Learn

Exercise Workbook

Version 2.0

Neil Silvers

Table of Contents

on

Introduction

This training is designed to provide the student with hands-on learning in troubleshooting common software issues. These issues range from install issues to major software crashes. The student will learn to use various tools in their efforts to better understand each situation. With the use of the Crash & Learn utility the student will be able to reproduce various software errors in a controlled environment. The exercises should be completed in the order listed.

The hands-on exercises in this manual include:

1 Installation

2 Software Errors or Traps

3 Runtime Errors

4 General Protection Faults

5 Visual C Runtime Errors

6 System Hangs and Lockups

7 System Crashes AKA BSOD

Exercise 1: Installation

Exercise Overview

This exercise will cover install and uninstall logging. Many software issues begin during the installation phase. Some of these install issues are obvious such as installing software designed for a Windows XP Operating System onto a Windows 95 Operating System. Other issues may require a deeper analysis. The goal of this module is to introduce the student to installation logging techniques. These logs will assist the student in analyzing the installation procedure.

Exercise is not available in Sample

Summary

At this point you have learned how to analyze log files from the install and the uninstall phases. This information will help you resolve common software issues. In the case where you may need escalated help, the ability to produce these log files will assist other software professionals in analyzing the issue.

Exercise 2: Software Errors or Traps

Exercise Overview

This exercise will cover the use of the Crash & Learn program in producing software errors and error traps.

Steps

1. Install Crash & Learn.
2. Review the Crash and Learn Readm.doc after installation.
3. After Crash & Learn is installed double click the crash icon on the system desktop.
4. In reviewing the Crash & Learn utility you will notice as you move your mouse over the row of buttons that different error types displayed.
5. On the right side of the UI window you will see that you can use Demo Mode or Real Mode.
6. Demo Mode will produce screen shots of the real error and does not produce the real error.
7. Select Real Mode.
8. Click on the first button to create a software error.
9. Answer yes to create the error.
10. The box above the button must contain a character in order to avoid the software error.
11. Add any letter to the box and try the operation again.
12. Now that you have resolved the software error, let's discuss how this error is used.

Most programmers will add a software trap to their code when a field is left blank. This could be a form that has required fields that can not be left blank. This type of error is created by design. In some cases if this field is left blank and the programmer did not use a software trap a different error could occur that has greater consequences. We will discuss those issues later.

Summary

At this point you have learned about the most common of software errors. The software trap is more of a design by the programmer to prompt the user to enter data or perform a certain task.

Exercise 3: Runtime Errors

Exercise Overview

This exercise will introduce a runtime error that can occur while running an application. It will add to the information already discussed in the previous exercise about software traps.

Steps

1. Launch the Crash & Learn application.
2. Select Real Mode.
3. Verify the data box above the first button is empty.
4. Click on the second button to create a runtime error.
5. Answer yes to create the error.
6. A Runtime Error 13 should display.
7. Look up the runtime error in Appendix B.
8. Launch the Crash & Learn application.
9. Select Real Mode.
10. Enter a zero into the data box.
11. Click on the second button to create a runtime error.
12. Answer yes to create the error.
13. A Runtime Error 11 should be displayed.
14. Look up the runtime error in Appendix B.
15. Launch the Crash & Learn application.
16. Select Real Mode.
17. Add a number greater than zero to the data box.
18. Click on the second button to create a runtime error.
19. Answer yes to create the error.

A message displays indicating that the problem has been resolved.

Summary

In the previous exercise you learned how software traps designed by programmers will generate errors and how these errors control the application. With the runtime errors generated in this exercise the errors were not controlled and the application closed. Users of programs will always enter incorrect data at various times that can not be controlled by the programmer. However adding software traps to the program can prevent the application from crashing.

Exercise 4: Other Runtime Errors

Exercise Overview

There are many types of runtime errors. In this section we will explore other types of runtime errors and look at Dynamic Link Library (DLL) files.

Steps

Exercise is not available in Sample

Summary

There are many causes for a runtime error. If the error message does list a specific file or DLL contact the software provider for a newer file. Also use the Internet for information about a specific file. In some cases other users of the same program may have experience the same issue. But be cautious of where the files come from. In some cases the files from unsupported sources may contain a virus.

Exercise 5: Visual C Runtime Errors

Exercise Overview

This exercise will cover another type of runtime error and these are created from the Visual C++ Library.

Steps

Exercise is not available in Sample

Summary

Some errors are so serious that the Operation System will keep a log of the failure. Using the event viewer to gather information about the failure is very useful. It will tell you when the failure occurred along with the application involved and the module of that application. In some case the information listed in the original error, in this case dbgheep.c, is not the cause of the error. Using the event viewer provided the correct information to resolve the issue.

Exercise 6: General Protection Faults

Exercise Overview

This exercise will cover the creation of a General Protection Fault.

Steps

Exercise is not available in Sample

Summary

Dr. Watson is another powerful tool available to help you analyze your errors. To learn more about Dr. Watson click on the help button in that program. Change the crash mode type from mini to full. Do you see a difference? Enable Dump Symbol table, what information was added?

Under Vista you have found another tool to use along with the Event Viewer.

Exercise 7: System Hangs and Lockups

Exercise Overview

This exercise will cover System Hangs and Lockups. This testing will require the student to power off their system, so it is recommended that all other applications be closed so data is lost or corrupted.

Steps

Exercise is not available in Sample

Summary

System lockups can be a big issue. If you have any open documents at the time of the lockup data can be lost or damaged. The other issue as you have seen is that no report from Windows application is generated. This can make resolving the issue very difficult. Using debugging tools such as WinDbg is very important.

Exercise 8: System Crashes AKA BSOD

Exercise Overview

This exercise will cover System Crashes also know as a BSOD (Blue Screen of Death). This testing will force the system to shut down, so it is recommended that all other applications be closed so data is lost or corrupted.

Steps

Exercise is not available in Sample

Summary

A system crash is often considered to be fatal, but as you have seen a system lock up can be worst because no report of the issue is generated. In the Windows folder you will see a file called MEMORY.DMP. This file contains information about your system and what may have caused the crash. Using tools such as WinDbg will help the engineers to help analyze the file.

Conclusion

At this point you have been exposed to all the major features and functions of the Crash & Learn utility. Also, you have been exposed to various tools that will help you analyze various software issues.

It is a good, solid foundation to build upon for troubleshooting software is an art and the environment is always evolving. Everyday new operating systems are introduced along with new tools to assist in troubleshooting. As a software professional you will meet new and exciting challenges everyday.

APPENDIX B
Runtime Errors

When a program is running, or executing, it is said to be in runtime. The term is mostly used by software developers to specify when errors in a program occur. A "runtime error" is an error that happens while the program is executing.

This is in comparison to a program that does not compile or assemble. All of the pieces of the program were assembled correctly but the code was not able to handle the command correctly. Or in some cases what was expected as input was not received correctly. A common runtime error is a divided by zero error. The expected input was a number that can be divided by zero but the input received was a zero or an alpha character

Below is a partial list of common runtime errors.

Code	Description
5	Invalid procedure call or argument
6	Overflow
7	Out of memory
9	Subscript out of range
10	This array is fixed or temporarily locked
11	Division by zero
13	Type mismatch
14	Out of string space
17	Cannot perform requested operation
28	Out of stack space
35	Sub or Function not defined
48	Error in loading DLL
51	Internal error
52	Bad file name or number
53	File not found
54	Bad file mode
55	File already open
57	Device I/O error
58	File already exists
61	Disk full
62	Input past end of file
67	Too many files
68	Device unavailable
70	Permission denied
71	Disk not ready
74	Cannot rename with different drive
75	Path/File access error
76	Path not found
91	Object variable or With block variable not set
92	For loop not initialized
93	Invalid pattern string

94 Invalid use of Null
322 Cannot create necessary temporary file
424 Object required
429 Automation server cannot create object
430 Class does not support automation
432 File name or class name not found during automation operation
438 Object does not support property or method <item>
440 Automation error
445 Object does not support this action
446 Object does not support named arguments
447 Object does not support current locale setting
448 Named argument not found
449 Argument not optional
450 Wrong number of arguments or invalid property assignment
451 Object not a collection
453 Specified DLL function not found
458 Variable uses an automation type not supported in JScript
501 Cannot assign to variable
502 Object not safe for scripting
503 Object not safe for initializing
504 Object not safe for creating
5000 Cannot assign to 'this'
5001 <Item> is not a number; Number expected
5002 <Item> is not a function; Function expected
5003 Cannot assign to a function result
5004 <Item> is not an indexable object; Cannot index object
5005 <Item> is not a string; String expected
5006 <Item> is not a date object; Date object expected
5007 <Item> is not an object; Object expected
5008 Cannot assign to <item>; Illegal assignment
5009 <Item> is undefined; Undefined identifier
5010 <Item> is not a boolean; Boolean expected
5011 Cannot execute code from a freed script
5012 Cannot delete <item>; Object member expected
5013 <Item> is not a VBArray; VBArray expected
5014 <Item> is not a JScript object; JScript object expected
5015 <Item> is not an enumerator object; Enumerator object expected
5016 <Item> is not a regular expression object; Regular Expression object expected
5017 Syntax error in regular expression
5018 Unexpected quantifier
5019 Expected ']' in regular expression
5020 Expected ')' in regular expression
5021 Invalid range in character set
5022 Expected '[' in regular expression
5023 Expected '(' in regular expression

6000 Variable is undefined or is not an object
6001 Method undefined or not found