

VT Advantage System Crashes on the Install PC

Document ID: 50842

Introduction

Prerequisites

- Requirements

- Components Used

- Conventions

Set up Your PC to Capture a System Crash

- Use the Microsoft Debugger

- Configure the Debugger

- Debug the Output File

Data Analysis

- Example of Trace where Cisco VT Advantage is not at Fault for the PC Crash

- Use the Data to Resolve the Crash

Related Information

Introduction

This document discusses what to do if you see Blue Screen crashes when you use or install Cisco VT Advantage. Some of these crashes result from possible software bugs in the Cisco VT Advantage drivers and others are from defects in other software products. This document helps you gather trace information which helps analyze the root cause for the crash.

Prerequisites

Requirements

Readers of this document need to be familiar with how to troubleshoot PC related issues.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

Refer to the Cisco Technical Tips Conventions for more information on document conventions.

Set up Your PC to Capture a System Crash

In order to properly catch a Blue Screen of Death (BSOD), configure your PC to issue a complete memory dump. The Cisco VT Advantage install program currently does this automatically. You can skip over these instructions if you have Cisco VT Advantage installed.

1. Right-click on the **My Computer** icon and select **Properties** on Windows XP or 2000 to configure your memory dump.
2. Click the Advanced tab in the window that appears.
3. Click either **Settings** (for Windows XP) or **Startup and Recovery** (for Windows 2000) in the Startup and Recovery section.

The Startup and Recovery dialog box appears.

4. Select the **Complete memory dump** option for that control in the Write debugging information drop-down list (towards the bottom).

Once the computer crashes, locate the memory.dmp file that was created and view it with the Cisco VT Advantage Error Reporting tool. If you do this yourself, navigate to the Windows directory (usually located on the C: drive) and locate memory.dmp.

If an end user performs this action, Cisco recommends that they use the Cisco VT Advantage Error Reporting tool. This makes locating the file much quicker. The error reporting tool zips the file and stores the result in a .zip file on the desktop of the user.

Use the Microsoft Debugger

The memory.dmp file is a raw image of the computer's memory at the exact moment the crash occurred. You cannot load this file into an editor or any other conventional resources. Instead, you need a copy of Microsoft's WinDbg debugger, which you can get from Microsoft . Download the most recent version of the NT-compatible tool set.

Note: Do not get the Windows Me/98 series tools by mistake.

After you download the program, go through the wizard to install the program on your computer. After it is installed you can see the program in your Start up menu as "Debugging Tools for Windows".



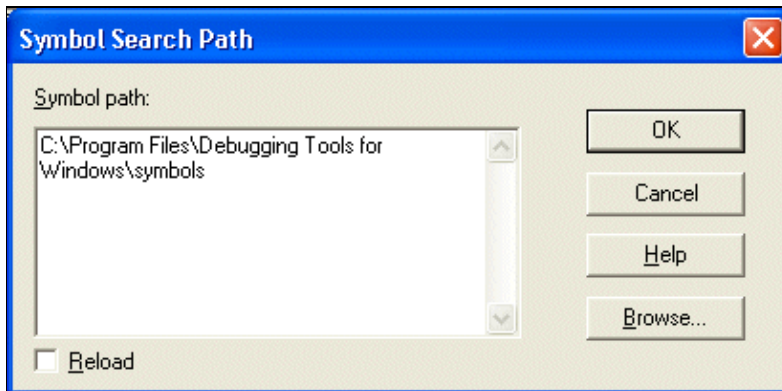
Configure the Debugger

Before you can use the debugger, you need to configure it to use Microsoft's online symbol server. This is a very simple process. This is the step-by-step configuration procedure.

1. Select **Start > Debugging Tools for Windows > WinDbg** to invoke the debugger.
2. From the File menu item, select **Symbol File Path** which brings up a dialog that contains a blank text field.

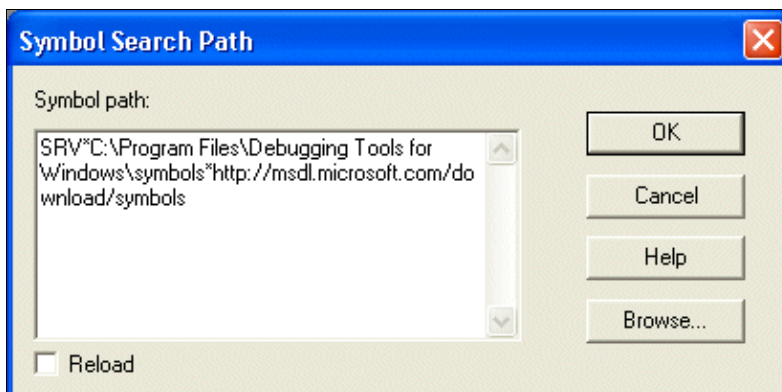
3. Browse to a directory that you plan to use as a storage area for symbol files.

In this example, the symbol files are placed in a new folder called 'symbols' in the same place as the application itself. A new folder is also created from the browser dialog. After you create the folder and select it, the dialog box appears similar to this image.



4. In the same Symbol Search Path text box, you need to change the string slightly.

With Microsoft's inscrutable terminology, add the string **SRV*** in front of the path, and the string ***http://msdl.microsoft.com/download/symbols** after it. When you are done, the text box appears similar to this.

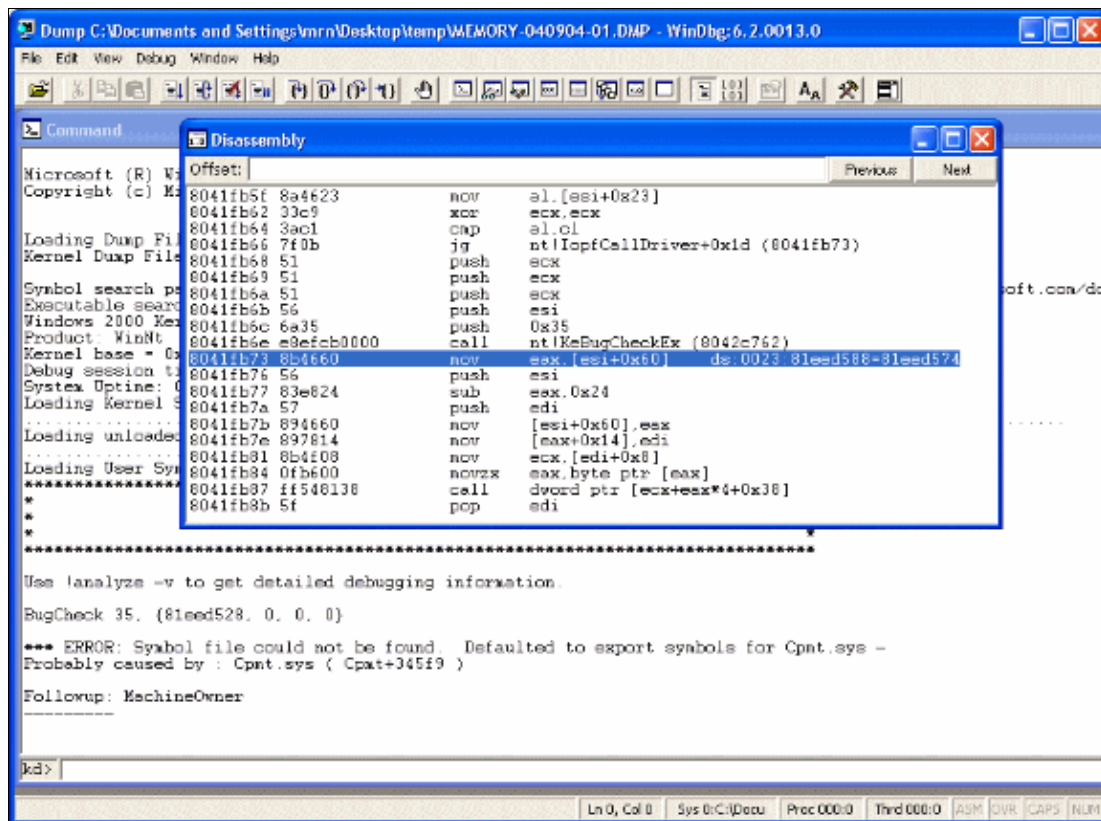


5. After you have this in place, select **Save Workspace** from the File menu.

In order to verify that the install and configuration are successful, verify that you see directories of symbol data present in the folder you just created after a debugging session.

Debug the Output File

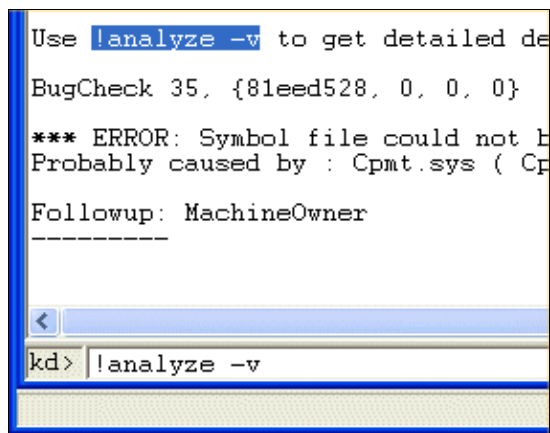
Invoke the debugger using the process described in Configure the Debugger. You can now drag the memory.dmp file directly into the main window, or open it with the use of the Open Crash Dump entry from the file menu. Either way, when the WinDbg opens the file, it probably needs to download symbol files from the Microsoft server. You might experience a 15 or 30 second pause. Eventually, a screen similar to this appears:



At this point, the debugger has a fair amount of information about the crash. However, you can perform one additional step to gather more information. Close the Disassembly window. In the command pane, there is a line that tells you this:

Use !analyze -v to get detailed debugging information.

Cut and paste this short command into the prompt line at the very bottom of the screen:



The debugger processes the command and produces the analysis of the crash in the command window:

```

Dump C:\Documents and Settings\mrm\Desktop\temp\MEMORY_040904-01.DMP - WinDbg: 6.2.0013.0
File Edit View Debug Window Help
Command
STACK_TEXT:
80473940 b001c5f9 8142c200 000005b2 81eec968 nt!IoPfcallDriver+0x1d
WARNING: Stack unwind information not available. Following frames may be wrong.
80473958 b001baf1 81eec968 00000000 00000000 Cpmt+0x345f9
804739e4 b0015683 81e3a328 81e3a0e0 80473b40 Cpmt+0x33af1
804739c0 b0015485 81e3b6a4 81e3a0e0 80000205 Cpmt+0x2d683
804739f0 b0015408 00000000 80473a50 00000000 Cpmt+0x2d485
80473a14 b000b4d9 81e3a0e0 3f2943c8 01c41dd5 Cpmt+0x2d408
80473b24 80431f81 8046f930 8046fbc0 ffdff000 Cpmt+0x234d9
80473b50 80464bd4 80482840 00000000 0014741e nt!KiTimerExpiration+0xb4
80473b64 80464b6f 0000000e 00000000 00000000 nt!KiRetireDpcList+0x30
80473b6c 00000000 00000000 00000000 00000000 nt!KiIdleLoop+0x26

FOLLOWUP_IP:
Cpmt+345f9
b001c5f9 8d8e84000000 lea ecx,[esi+0x84]

FOLLOWUP_NAME: MachineOwner

SYMBOL_NAME: Cpmt+345f9

MODULE_NAME: Cpmt

IMAGE_NAME: Cpmt.sys

DEBUG_FLR_IMAGE_TIMESTAMP: 4023f9f5

STACK_COMMAND: kb

BUCKET_ID: 0x35_Cpmt+345f9

Followup: MachineOwner

kd>
Ln 0, Col 0 Sys 0:C:\Docu Proc 000:0 Thrd 000:0 ADW DWR CAPS NUR

```

Data Analysis

At this point, you can begin to determine where the crash occurred. The debugger has printed a stack trace that shows where the error occurred, and the context in which it happened.

In the command window, some off-screen text provides the source of the problem with this analysis:

```

NO_MORE_IRP_STACK_LOCATIONS (35)
A higher level driver has attempted to call a lower level driver through
the IoCallDriver() interface, but there are no more stack locations in the
packet, hence, the lower level driver would not be able to access its
parameters, as there are no parameters for it

```

Find out which driver caused the problem. This information is in the stack trace:

```

STACK_TEXT:
80473940 b001c5f9 8142c200 000005b2 81eec968 nt!IoPfcallDriver+0x1d
WARNING: Stack unwind information not available. Following frames may be wrong.
80473958 b001baf1 81eec968 00000000 00000000 Cpmt+0x345f9
804739e4 b0015683 81e3a328 81e3a0e0 80473b40 Cpmt+0x33af1
804739c0 b0015485 81e3b6a4 81e3a0e0 80000205 Cpmt+0x2d683
804739f0 b0015408 00000000 80473a50 00000000 Cpmt+0x2d485
80473a14 b000b4d9 81e3a0e0 3f2943c8 01c41dd5 Cpmt+0x2d408
80473b24 80431f81 8046f930 8046fbc0 ffdff000 Cpmt+0x234d9
80473b50 80464bd4 80482840 00000000 0014741e nt!KiTimerExpiration+0xb4
80473b64 80464b6f 0000000e 00000000 00000000 nt!KiRetireDpcList+0x30
80473b6c 00000000 00000000 00000000 00000000 nt!KiIdleLoop+0x26

```

This shows that the error occurred in an NT internal routine that was called by code at least five functions deep in the CPMT driver. The CPMT driver is the Cisco driver. At this point it is possible that this dump needs to go to Cisco Technical Support for more analysis.

This list shows the names of the drivers that ship with Cisco VT Advantage:

- **Cpmt** Cisco Media Termination driver.
- **CamDrC21.sys** USB Camera driver for Windows XP.
- **CamDrC20.sys** USB Camera driver for Windows 2000.
- **CamUsC20.sys** USB Camera driver for Windows 2000.
- **CdpPacketWdmCvl.sys** CDP Packet driver.

If you see any of these names in the stack trace from a BSOD, it is likely that Cisco VT Advantage is involved in the crash. If you do not see any of the Cisco drivers, you can start to look at other drivers as the possible cause.

Example of Trace where Cisco VT Advantage is not at Fault for the PC Crash

Often when the PC crashes while it uses Cisco VT Advantage, the Cisco product is absolved of responsibility for the crash. This is an example of a stack trace that was pulled out of a BSOD memory dump after a crash where Cisco VT Advantage was installed on the PC:

```
STACK_TEXT:  
WARNING: Stack unwind information not available. Following frames may be wrong.  
804739f0 bfd7539b 80473a74 00000000 8204e0d8 SynTP+0x630c  
80473a98 bfd74d9c 00000001 8204ee50 00000002 SynTP+0x539b  
80473ab8 bfd84a42 00000001 0000003d 81fdb000 SynTP+0x4d9c  
80473ad8 bfd85755 846db868 846db880 80473b48 SynTP+0x14a42  
80473aec eb49715c 81fdb000 846db868 846db880 SynTP+0x15755  
80473b50 80464bd4 8204e210 8204e001 00000000 i8042prt!I8xMouseQueryWmiDataBlock+0x80  
80473b64 80464b6f 0000000e 00000000 00000000 nt!KiRetireDpcList+0x30  
80473b6c 00000000 00000000 00000000 00000000 nt!KiIdleLoop+0x26
```

Here you can see that the CPU is several function calls deep into a driver called SynTP (which is a TouchPad driver). Combined with a little bit of analysis on additional lines in the command window, the conclusion is that this driver is what caused the problem and there could be a bug.

How do you verify this?

This list shows the names of the drivers that ship with Cisco VT Advantage:

- **Cpmt** Cisco Media Termination driver.
- **CamDrC21.sys** USB Camera driver for Windows XP.
- **CamDrC20.sys** USB Camera driver for Windows 2000.
- **CamUsC20.sys** USB Camera driver for Windows 2000.
- **CdpPacketWdmCvl.sys** CDP Packet driver.

If you see any of these names in the stack trace from a BSOD, it is likely that the Cisco driver was involved in the crash. If you do not see any of the Cisco drivers, you can start to look at other software programs.

Use the Data to Resolve the Crash

In one example, a user receives a BSOD crash every time they dock while Cisco VT Advantage runs. After you review the crash output, you can look at the file and point out that the crash happens in the driver for the network card of the PC. Once the Cisco driver is installed and is in use, the runtime environment changes. Also the Cisco driver allocates memory, taking CPU time slices, and many other types of resources. Other drivers can suddenly find that they cannot allocate as much memory as they wish, or there is a delay when

they respond to events. This can result in a system crash. Most drivers should be robust enough to handle this reallocation of resources.

If the failure is determined to be drivers of other vendor products, go to the web site of the vendor and download the latest drivers. Often issues like this are fixed with patches that are available on the vendor web site.

Related Information

- **Voice Technology Support**
- **Voice and IP Communications Product Support**
- **Recommended Reading: Troubleshooting Cisco IP Telephony**
- **Technical Support & Documentation – Cisco Systems**

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Nov 16, 2005

Document ID: 50842
