The Blue Screen Of Death (BSOD) Primer

By SunPCi Product Development Team

http://www.sun.com/desktop/products/sunpci/

Introduction

If you have ever used a Microsoft based Windows system (aka., Wintel), then you have also have experienced, at one point or another, the joys and terrors of ***"The Fatal Exception",*** lovingly called the "Blue Screen Of Death", or BSOD. The intent of this paper is to provide some explanation, expose some of the "kernel" details on what a BSOD means and possibly give some remedial action that may help in recovery and/or diagnosis. Basically, we try and give the reader a bit of a "secret decoder ring" for these banes of driver writers. However, note that in no way is this intended to be a complete document on the interpretation of BSODs, but this is merely a compendium of "interesting information" about BSODs which have appeared in our development, debugging and QA efforts for SunPCi™ software.

Note: If the Windows system is configured to automatically restart after a system failure, the user will **not** see a BSOD, but only notice that the SunPCi session resets. The user should insure that this feature is disabled. To disable this feature, this is typically found in the *SystemProperties* dialog. For more details on disabling this feature, please reference your Microsoft documentation.

What's a BSOD?

When Windows encounters a condition that compromises safe system operation (i.e., a "bug"), the system halts. This condition is called a *bug check*. It is also commonly referred to as a *system crash*, a *kernel error*, a *system fault,* or a *Stop error*. This is analogous to a panic in Solaris Operating Environment. The screen will switch into VGAtext mode, draw a dark blue background and then display an error message. This blue color, and the fact that the system has now stopped gives rise to the term the Blue Screen of Death (BSOD).

The exact appearance of the BSOD message depends upon the cause of the error. The BSOD could take the form:

STOP : 0x00000079 (0x00000002, 0x00000001, 0x00000002, 0x00000000)

In this case, the first value (**0x00000079 -** all values in this document are hexadecimal) is known as the *bugcode* or *Stop code*. The other 4 values (within the parentheses) are the BSOD's parameter list and have values that are dependent upon the bugcode.

Another BSOD presentation variation is:

STOP : c000021a (Fatal System Error).

Depending upon the bugcode, other textual information may be displayed on the blue screen. This can include:

- The name of the offending driver, service or facility
- Explanatory text of the reason for the BSOD
- Possible recovery techniques
- Kernel stack trace, with addresses

There are over 250 "documented" BSOD codes, and many of these are documented with "This bug check appears very infrequently." (this is the full text quoted directly from the Microsoft documentation). And, unfortunately, most BSOD codes contain addresses, which, unless the Microsoft debugging tool WinDBG is attached to the running system, remain of little use. Thus, many of the BSODs impart only a hint of the actual causes, and a much deeper investigation, with WinDBG and debug versions of drivers, must be initiated.

BSODs are primarily used for the debug of device drivers, and the majority only occur during the development phase of driver generation. The Microsoft Device Driver Kit (also known as the DDK, normally supplied with an MSDN subscription) contains a listing of all the "documented" BSOD

bugcodes. Note, however, that there are more bugcodes than are even documented in the DDK, and even more than can be found in the DDK header files.

Within the Microsoft driver writing community, there is a "most notorious felon" list of sources that tend to cause BSODs. This list of offenders is common to PCs in general and not just the SunPCi package. Thus, if you have added any of these types of systems, they might be the first to suspect for root causes of a BSOD. In general, the user should be wary of 3rd party hardware and drivers that have been installed which have not passed Microsoft's Hardware Compatibility Test (HCT) and therefore not a member of Microsoft's Hardware Compatibility List (HCL). This list includes:

- External CDROM/CD Writers
- External hard drives
- Antivirus scanners
- External backup devices
- Graphics mirroring software

What about Win9x BSODs?

This paper only deals with BSODs which occur while running Windows NT4, Windows 2000, Windows XP and .NET. While BSODs have occurred in Windows OS's since the days of Windows 3.0 (including Windows 3.11, Win95, Win98 and Win98 variations), the Windows kernel architecture underwent a significant change between Win9x and Windows NT4. It was first in Windows NT4 that this type of fault handling was standardized among all Windows OS facilities. Thus, the BSODs in Windows OS's prior to Windows NT4 are significantly different from those after Windows NT4.

If a BSOD occurs....

Well, you're out of luck. Windows has crashed and is not going any further until you take some action (like reboot/restart the SunPCi session). In all cases, there is no immediate recovery from a crash (that is, you can't immediately continue operation). One merely hopes that the data on the emulated disk has not been corrupted to such an extent that booting (or data retrieval) is not possible. Fortunately, the vast majority of cases do not result in disk or data corruption. This is especially true if the file system used is NTFS rather than FAT.

If a bugcheck occurs and is reproducible, the user should try to determine the steps to reproduce the crash, and, if possible determine what software package and/or which driver seems to be the culprit. If the BSOD contains a stack trace, the driver at the top of the stack listing is a likely suspect. Further, there are BSODs which may indicate specifically which driver is at fault.

In many cases, the bugcode and the parameter values are stored also in the SunPCi session logfile. However, there are cases where the BSOD occurs at a point in the operation of Windows where saving this information to the SunPCi log file is impossible. Therefore, the user is asked to always try and retain as much of the BSOD information as possible, either by transcribing it or by saving the SunPCi session display (in this case, the BSOD itself) as a screen shot (using either the Solaris Operating Environment tool *snapshot* or the X Windows tool *xwd)*.

Check the SunPCi log file for any messages that may indicate a SunPCi driver that may have failed or caused a failing condition. A critical part of debugging a BSOD occurrence is the retention of the SunPCi log file and submitting it with a SunPCi bug report. In general, filing a SunPCi bug report would only be helpful to the SunPCi Engineering staff if the BSOD is reliably reproducible. To insure maximum information transfer, the following should be submitted:

- SunPCi Bug Report
- SunPCi Session Log File

- BSOD transcription or screen shot
- List of all applications/drivers/hardware that have been installed and attached
- Specific list of steps/actions which causes the BSOD
- If possible, the actual C: drive on which the BSOD is reproducible

Note: Whenever a SunPCi session is started, the pre-existing SunPCi log file will be renamed to a file with the *.old* appended to the filename. Thus, the user must take care to preserve the SunPCi log file which contains the BSOD session information, otherwise, after 2 SunPCi sessions restarts, the BSOD SunPCi log file will be lost. For more information on SunPCi log files, see the SunPCi White Paper "**SunPCi Logging"**.

For more information on submitting a SunPCi Bug Report, see the SunPCi white paper "**How to Collect Useful SunPCi Bug Information**".

General Recovery Procedures

In general, there are not too many options for any type of recovery. Typically, one tries to just "reboot" the SunPCi hardware in the hopes that the BSOD occurred because of a rare condition of some driver which was overlooked in coding and testing. However, if the BSOD persists, there are some tactics that may be employed to repair the system and get the SunPCi session back to a state where the BSOD no longer occurs.

- If you had recently installed hardware or software, try removing or disabling it and see if the condition remains. Specifically, the user should be suspicious of 3rd party hardware/software that has not passed the Microsoft Hardware Compatibility Tests (HCT) and therefore not on the Microsoft Hardware Compatibility List (HCL). Additionally, drivers that are not digitally signed may be suspect.
- Check the System Log in the Event Viewer for messages that may identify the device or driver that has failed.
- After reboot, the user is suggested to check the disk integrity by running ***Chkdsk /f /r*** on all emulated drive partitions.

  Note: If you system partition is formatted with FAT (FAT16 or FAT32), the long filenames used by Windows can be damaged if ***scandisk*** or another MS-DOS based hard disk tools is used to verify the integrity of your hard disk from an MS-DOS prompt. Always use the version of ***Chkdsk*** that matches your Windows version. This is typically found `C:\winnt\system32` or `%SystemRoot%\system32`.

New Hardware Added/Existing Hardware Modified

This case seems to be one of the major causes of BSODs in a previously stable installation. If new hardware has been added, or existing hardware modified:

- Run manufacturer supplied diagnostics on the device.
- Make sure all adapter cards, memory sims, connectors are properly seated and fastened. Microsoft recommends to "use an ink eraser or an electrical contact treatment to ensure that all contacts are clean". However, it should be noted that ink erasers are not recommended as the gold on the card fingers is so thin that one may remove some/all of the metal, which may cause additional problems/failures.
- Make sure the latest drivers from the supplier have been installed and that they match the version and service pack of the Windows operating system.

- Check the **System Log** in the **Event Viewer** for any driver/service failures or errors.
- Check the SunPCi log file for any driver failure or errors.
- Remove and/or disable the drivers for that hardware and reboot to see if the error condition persists.

Safe Mode Recovery (Windows 2000, Windows XP and .NET only)

When booting a SunPCi session, a black screen will appear to indicate that the SunPCi hardware is going through its Power On Self Test (POST). It is at this point the user will see a screen that states the type of processor and its speed. After that, a list of Operating System choices will be displayed. Here, the user may be able to tell Windows to boot into Safe Mode, by depressing F8 when the SunPCi session displays the operating system choices.

If the BSOD had occurred after the installation of a new or updated device driver, the driver should be removed or replaced. If the error occurs during the startup sequence, you may be able to use Safe Mode booting to rename and/or delete the faulty driver.

Also, try restarting your SunPCi session by selecting **Advanced Options** menu and choose the **Last Known Good Configuration**. This was the setting of Windows used the last time the SunPCi session successfully booted. This option is most effective when only one driver or service is added at a time.

Driver Identification.

In some cases, the BSOD display will output the name of the driver which is causing the problem. If the named driver is not supplied by the SunPCi package and not supplied by Microsoft (thus, the driver is from a 3rd party manufacturer), then the user should try and remove (or disable) that driver from the system by uninstalling the driver. However, to know if the driver is from the SunPCi package or from Microsoft, and to aid in the analysis of the BSOD, the following tables list the names/identifiers of some of the major SunPCi and Microsoft drivers. This list of driver names, while certainly not complete nor exhaustive, has been provided to aid in the identification of a faulty driver, in the cases where neither the SunPCi Wintel drivers nor the base Windows operating system drivers are identified in the BSOD.

Note that as with all filenames in Windows, these are case insensitive.

SunPCi Driver Names

The following is a list of the most important SunPCi Wintel drivers supplied with the SunPCi software distribution:

Driver Name	Function
SunVideo.dll	Video (GDI) Display Driver
SunBrdge.sys	SPARC/Intel bridge driver (WinXP, .NET)
Bridge.sys	SPARC/Intel bridge driver (NT4, Win2K)
SunFSD.sys	Network File System Driver
SunEmdk.sys	Emulated Disk Driver (WinXP, .NET)
Emdisk.sys	Emulated Disk Driver (NT4, Win2K)
SunCdrom.sys	CDRom driver (WinXP, .NET)
SunFlppy.sys	Floppy driver (Win2K, WinXP, .NET)
SunNDIS.sys	NDIS driver
Sunnpdll.dll	Network provider
Sunppnt.dll	Print provider
Vidbcall.sys	Video Support Driver
Sunvmini.sys	Video Miniport Driver

Microsoft Driver Names

The following is a list of the most common NT drivers that may appear in a BSOD:

Driver Name	Function
NtosKrnl.exe	NT kernel
NTdll.dll	NT support library
Win32k.sys	Graphics Display Interface (GDI) Driver
Hal.dll	Hardware Abstraction Library

Kernel Terminology and Background

Before we describe the individual BSODs, there are some definitions of terminology that should be helpful, as they are used in the description of the BSODs.

Term	Definition
Bugcode	The hexadecimal value which identifies a BSOD. An example is 0xA, which identifies this BSOD as the IRQL_NOT_LESS_OR_EQUAL BOSD
Driver	This is a kernel mode program which is part of the Windows operating system which handles all requests to a particular device. An example would be the display driver, which takes commands from a windows application and directs the hardware on what to draw on the screen.
Exception	An error condition in a driver which causes a fault or system crash.
HCT	Hardware Compatibility Test. This is the very extensive Microsoft validation test for drivers that, if passed, qualify a driver to be placed on Microsoft's Hardware Compatibility List. If a driver successfully passes HCT, it is highly unlikely that there are many conditions which would cause the driver to spawn a BSOD. See the Microsoft web site http://www.microsoft.com/hwdq/hwtest/ for more details on the HCT.
HCL	Hardware Compatibility List. This is Microsoft's list of certified drivers that have passed the Hardware Compatibility Test (HCT). See the Microsoft web site http://www.microsoft.com/hwdq/hwtest/ for more details on the HCL.
IRQL	Interrupt ReQuest Level. While operating, the NT kernel will transition to and from various states. These states are known as IRQL and are identified by a set of increasing integers, from 0 to 31. At each IRQL there are specific rules that must be observed. An example would be that certain memory references/accesses can only be done at certain IRQLs and not at others. It is the drivers responsibility to know, at all times, its current IRQL at and what operation are valid at that IRQL Violation of any of these rules will cause a BSOD.
IRP	I/O Request Packet. This is a kernel structure which is passed between drivers to communicate an I/O request. A typical I/O request would be actions like Read, Write, Open, Close, IOCTL, etc.
NonPaged Pool	This is an area of kernel memory that cannot be paged out (e.g., Paged Pool). Typically, a driver will allocate such memory so that it may access it at any IRQL.
NT kernel	The generic name for the Windows operating systems after Win3x/Win9x. This includes Windows NT 4, Windows 2000, Windows XP and Windows .NET
Paged Pool	This is an area of kernel memory which can be paged out to disk if it is not currently being used or accessed.

Service or System Service	This is a program which is not part of the kernel (it is a user mode application) which performs requests (i.e., services) on behalf of other processes. That is, the user will never directly interact with a service, but the user's programs/applications may make requests which will be directed to a service for execution. An example of this is the print service, which handles all print request and printer manipulations. Typically, a service interacts with a particular kernel driver in a very specific fashion.
WinDBG	This is the Windows Kernel Mode DeBuGger that is supplied in the Microsoft Device Driver Kit (DDK). It is akin to an advanced version of UNIX/Solaris Operating Environment's dbx/gdb/kdb/kgdb.
Wintel	Shorthand for Windows/Intel

BSOD Descriptions

The following section details the most commonly occurring BSODs that have been seen in SunPCi engineering. The vast majority of these have been seen during the development phases of the SunPCi Wintel device drivers. However, after the SunPCi drivers have been tested and hardened, most of the BSODs that occur in the SunPCi QA/Testing lab have been directly related to 3rd party hardware and software incompatibilities with the base Windows operating system, and, for the most part, also occur on standard PCs.

Each bugcode is listed by its bugcode value and the name that it is identified when the BSOD appears. Following that is a general description of what the bugcode means and what may cause the BSOD. A list of the parameters and their meaning is then presented. Note that the majority of the time, the values of the parameters will not be very helpful to the reader of this document, but is shown here to underscore the importance of capturing as much of the BSOD information as possible, as these values may be quite important to the driver manufacturer (or the SunPCi Engineering team if it is a SunPCi Wintel driver at fault). Following the parameter listing is a list of possible recovery and/or workaround suggestions.

Bugcode 0xA - IRQL_NOT_LESS_OR_EQUAL

This is a fairly common BSOD that occurs when a driver has illegally accessed a memory location while NT is operating at a specific IRQL. This is a driver coding error, akin to trying to access an invalid memory location.

Parameters:

1 - memory location that was referenced

2 - IRQL at time of reference

3 - 0 == read, 1 == write

4 - code addressed which referenced memory

Recovery/Workaround:

There is none. This is a fatal error and is a driver coding error.

Bugcode 0x1E - KERNEL_MODE_EXCEPTION_NOT_HANDLED

This indicates that a driver has generated an exception (a fault) that the error handler did not catch. This is a fairly common BSOD and is typically due to a coding error or an unhandled/mishandled edge condition. Most commonly, this occurs when dereferencing a bad memory address (e.g., trying to dereference memory address 0x0).

The SunPCi QA team has seen this when installing EZ CD Creator 5.0 from Roxio™ when no CD writer has been attached to the system. This has also been seen while changing display modes on the external monitor in the SunPCi II distribution.

Parameters:

1 - exception code

0x80000002 = unaligned data reference encountered

0x80000003 = a kernel breakpoint/ASSERT encountered

0xC0000005 = memory access violation occurred

2 - address of the exception

3 – parameter 0 of the exception

4 – parameter 1 of the exception

Recovery/Workaround:

- Insure that you have the update the Wintel drivers to the latest sets from the installed SunPCi package.
- If 3rd party software and/or hardware have been installed, insure that you have installed the latest drivers from those manufacturers.
- This error can occur after the first restart during Windows Setup, or after Setup is finished. A possible cause is a lack of disk space for installation. Try reinstalling with a larger emulated disk, or delete any unneeded temporary files, Internet cache files, application backup files, and .chk files.

Bugcode 0x22 - FILE_SYSTEM

This is a special BSOD that has been used to indication a failure in the SunPCi Wintel drivers. This bugcheck occurs when a SunPCi driver detects an error condition that should never occur in a properly running system, for which continued execution could result in possible data corruption, and for which there is no available software recovery path. The first parameter for this particular bugcode is a special encoding that is used by the SunPCi driver team to help identify the failure path.

Parameters:

1 - embedded module tag/line number

2 - unused

3 - unused

4 - unused

Recovery/Workaround:

There is no recovery or workaround.

Bugcode 0x2E - DATA_BUS_ERROR

This typically indicates a parity error in system memory has been detected and is almost always caused by a hardware problem, which being a configuration issue, defective or incompatible hardware. The most common causes are defective RAM, Level 2 RAM cache errors, or video RAM errors. Hard disk corruption can also cause this error. This may also be due to a driver coding error, accessing an invalid address.

Parameters:

1 - Virtual address that caused the fault

2 - Physical address that caused the fault

3 - Processor status register (PSR)

4 - Faulting instruction register (FIR)

Recovery/Workaround:

- This may be due to a hardware reconfiguration. Refer to the **New Hardware Added/Existing Hardware Modified** section in the **General Recovery Procedures** section.
- Check the system for viruses, using any up-to-date commercial virus scanning software that examines the Master Boot Record of the hard disk.
- Run **Chkdsk /f /r** on the system partition.

Bugcode 0x3F - NO_MORE_SYSTEM_PTES

This is the result of a system that has performed too many I/O actions and the system has a fragmented system page table (PTE). In actuality, the system is typically not out of PTEs, but a driver has requested a large block of memory, but there is no contiguous block of sufficient size to satisfy this request.

Parameters:

1 - 0 == system expansion PTE type

1 == nonpaged pool expansion PTE type

2 - size of memory request

3 - total free system PTEs

4 - total system PTEs

Recovery/Workaround:

- Remove any recently installed software, especially backup utilities or disk-intensive applications.

Bugcode 0x44 - MULTIPLE_IRP_COMPLETE_REQUESTS

This indicates a fault in driver logic (an I/O Request Packet (IRP) has been tagged as completed more than once). This has been seen to occur on a heavily loaded/stressed system. To aid in diagnosis, the user should note all applications that were executing at the time of the failure.

This has been seen in the SunPCi QA when mistakenly installing multiple copies of Roxio's EZ CD Creator.

Parameters:

1 - Address of the IRP

2 - Reserved

3 - Reserved

4 - Reserved

Recovery/Workaround:

There is no recovery or workaround.

Bugcode 0x4E - PFN_LIST_CORRUPT

This indicates a memory corruption in a driver (the Page Frame Number (PFN) list is corrupt). This has been seen to occur on a heavily loaded/stressed system. To aid in diagnosis, the user should note all applications that were executing at the time of the failure.

Parameters:

1 - type of corruption

2 - page frame or entry number at time of failure

3 - page information

4 - reserved

Recovery/Workaround:

There is no recovery or workaround.

Bugcode 0x5 - INVALID_PROCESS_ATTACH_ATTEMPT

This indicates that a kernel process was making an attempt to attach to another process. This has been seen to occur on windows server while running stress tests. To aid in diagnosis, the user should note all applications that were executing at the time of the failure.

Parameters:

1 - reserved

2 - reserved

3 - reserved

4 - reserved

Recovery/Workaround:

There is no recovery or workaround.

Bugcode 0x50 - PAGE_FAULT_IN_NONPAGED_AREA

This indicates that a driver fault where a driver has attempted to access memory that cannot be accessed in the driver's current IRQL. The system will try and identify the faulty driver by printing the name of the driver with the BSOD information.

Parameters:

1 - memory address referenced

2 - 0 == read operation

1 == write operation

3 - address that referenced memory

4 - reserved

Recovery/Workaround:

If the faulting driver has not been identified as a SunPCi driver, then there are 2 common possibilities

- This may be due to a hardware reconfiguration. Refer to the **New Hardware Added/Existing Hardware Modified** section in the**General Recovery Procedures** section.
- If an antivirus system has been installed, attempt to disable the antivirus scanning. If the problem persists, remove the antivirus program from the system.

Bugcode 0x7B - INACCESSIBLE_BOOT_DEVICE

This indicates that Windows can't access the system partition during setup/boot.

Parameters:

1 - reserved

2 - 0

3 - 0

4 - 0

Recovery/Workaround:

The vast majority of the cases of this BSOD are when the user attempts to install a windows operating system and does NOT follow the special SunPCi installation instructions. Instead, the user has attempted to install Windows using the Microsoft supplied installer (e.g., **setup.exe** or **winnt.exe** in the `I386` directory of the installation medium). The reason that this BSOD occurs is since the SunPCi product uses an emulated disk, there are special SunPCi drivers that must exist at Windows boot to read the information on the emulated disk. These drivers are not supplied on any Microsoft distribution. The recovery, in this case, is to restart the installation using the method outlined in the SunPCi documentation.

This BSOD has also been seen to occur at times other than at Windows installation. If Windows has successfully been installed and this BSOD occurs, the user should try to boot in safe mode and run **Chkdsk**. Otherwise, the emulated drive is corrupted beyond repair.

Bugcode 0x7F - UNEXPECTED_KERNEL_MODE_TRAP

This indicates that a trap (fault) was generated by the CPU and the kernel failed to catch this trap. Note that this could be a 1st or 2nd level trap, the latter being a fault that occurred while processing an earlier fault, which always results in a system crash.

Parameters:

1 - trap number

0x000000000 - Divide by Zero error

0x000000004 - Overflow

0x000000005 - Bounds check fault

0x000000006 - Invalid Opcode

0x000000008 - Double/Multiple faults

2 - reserved

3 - reserved

4 - reserved

Recovery/Workaround:

This may be due to a hardware reconfiguration. Refer to the **New Hardware Added/Existing Hardware Modified** section in the**General Recovery Procedures** section.

Bugcode 0x8E - KERNEL_MODE_EXCEPTION_NOT_HANDLED

This indicates that a driver has generated an exception (a fault) which the error handler did not catch. This is a fairly common BSOD and is typically due to a coding error or edge condition. Most commonly, this occurs when dereferencing a bad memory address (e.g., trying to dereference memory address 0x0).

Parameters:

1 - exception code

0x80000002 = unaligned data reference encountered

0x80000003 = a kernel breakpoint/ASSERT encountered

0xC0000005 = memory access violation occurred

2 - address of the exception vector

3 - trap frame

4 - reserved

Recovery/Workaround:

- Insure that you have updated the Wintel drivers with the latest sets of drivers from the SunPCi package as well as the latest drivers from other manufacturers.
- This error can occur after the first restart during Windows Setup, or after Setup is finished. A possible cause is a lack of disk space for installation. Try reinstalling with a larger emulated disk, or delete any unneeded temporary files, Internet cache files, application backup files, and .chk files.

Bugcode 0xA - IRQL_NOT_LESS_OR_EQUAL

This indicates that a driver has accessed memory at an inappropriate IRQL. This is a fairly common driver fault, whereby a memory address references a memory page which is currently paged out and the CPU is in an IRQL which does not allow that type of memory access. This usually occurs after the installation of a faulty device driver or system service.

If this is encountered while upgrading to a later version of Microsoft Windows, this error might be caused by a device driver, system service, virus scanner, or backup tool that is incompatible with the new (target) version.

Parameters:

1 - memory referenced

2 - IRQL at time of reference

3 - 0 == Read, 1 == Write

4 - Address which referenced memory

Recovery/Workaround:

- To resolve an error caused by a faulty device driver or system service, restart the computer, press F8 at the operating system choices and select **Last Known Good Configuration** from the **Advanced Options** menu and then remove the driver or service.
- To resolve an error caused by an incompatible device driver, system service, virus scanner or backup tool, check the system Log in the Event viewer that might identify the device or

driver that caused the error. Run diagnostics supplied by the driver manufacturer. Also, make sure that latest Service Pack is installed.

NOTE : Microsoft recommends that "before one upgrades to a new version of Windows, the user should remove all third-party device drivers and system services, and disable any virus scanners and then contact the software manufacturers to obtain updates of these third-party tools."

Bugcode 0xC000021A - STATUS_SYSTEM_PROCESS_TERMINATED

This indicates that an error has occurred in the graphics subsystem. Windows requires that the graphics subsystem properly executes in order for the rest of Windows to continue operation. Most times, the "faulty" driver will be identified as winlogon.exe. This has been seen as a result of the corruption of win32k.sys or SunVideo.dll. However, this corruption has not been a real file corruption, but rather the corruption of the transfer of the image information from the emulated disk (resident on the Solaris Operating System file system), to the Windows operating system. Furthermore, this even has only ever been seen (rarely) with the SunPCi II distribution running on Ultra™ 30 or Ultra 60 workstations.

Parameters:

1 - reserved

2 - reserved

3 - reserved

4 - reserved

Recovery/Workaround:

Typically, this is a transient and rare error and is almost always resolved by rebooting/restarting the SunPCi session, and was never an indication of any real disk corruption. However, if the error persists for a particular file, the disk may be corrupted. Run **Chkdsk /f /r** to detect and resolve any file system structural corruption.

Bugcode 0xC0000221 - STATUS_IMAGE_CHECKSUM_MISMATCH

This indicates that a driver or a system library has been corrupted on disk. This BSOD was only ever seen (rarely) with the SunPCi II distribution running on Ultra 30 and Ultra 60 workstations. Typically, this was an error in the transfer of image data from the emulated disk (resident on the Solaris Operating System file system) to the Windows Operating system.

Parameters:

1 - reserved

2 - reserved

3 - reserved

4 - reserved

Recovery/Workaround:

Typically, this is a transient and rare error and is almost always resolved by rebooting/restarting the SunPCi session, and was never an indication of any real disk corruption. However, if the error persists for a particular file, the disk may be corrupted. Run **Chkdsk /f /r** to detect and resolve any file system structural corruption.

Bugcheck 0xC2 - BAD_POOL_CALLER

This indicates that a kernel thread/process is making an invalid memory allocation request.

Parameters:

1 - reserved

2 - reserved

3 - reserved

4 - reserved

Recovery/Workaround:

There is no recovery/workaround for this BSOD. This is a fatal flaw in the driver.

Bugcheck 0xC4 - DRIVER_VERIFIER_DETECTED_VIOLATION

This indicates that the Driver Verifier facility has detected a violation of Microsoft driver guidelines/rules.

Parameters:

1 - error code

2 - reserved

3 - reserved

4 - reserved

Recovery/Workaround:

This BSOD only occurs if the Driver Verifier facility has been enabled. This facility should only be used by SunPCi driver developers and should not be active on production SunPCi sessions. The Driver Verifier is also automatically enabled when running Microsoft Hardware Compatibility Tests (HCTs). In any case, the Driver Verify should be disabled and the SunPCi session rebooted.

Bugcode 0xD1 - DRIVER_IRQL_NOT_LESS_OR_EQUAL

This indicates that a driver has attempted to access memory while the CPU is in an inappropriate IRQL.

Parameters:

1 - memory address referenced

2 - IRQL

3 - 0 == Read, 1 == Write

4 - address that referenced memory

Recovery/Workaround:

There is no recovery or workaround for this BSOD.

In Conclusion

You now have some background, and some possible remedies, for BSODs. Again, this document is not meant to be a complete guide, but a quick reference to BSODs that the SunPCi QA team has experienced while qualifying SunPCi software and hardware.

For more details concerning BSODs, consult the Microsoft Device Driver Kit (DDK) Documentation and/or the Microsoft Developer Network (MSDN) web site (http://www.msdn.microsoft.com). Additionally, there are a number of publications on Microsoft device drivers and system internals that contain more details about BSODs (again, consult the MSDN web site for a listing of publications).