

SunPCi™ Logging

By the SunPCi Product Development Team

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

Copyright 2002 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, SunPCi and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2002 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, SunPCi et Solaris sont des marques de fabrique, ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Introduction

It might come to some readers as a great surprise that not all computer software is perfect. That is, there are some rare cases where software may actually fail. The SunPCi™ software falls into this rather large category, such failures resulting in either a Solaris™ core dump or a Windows Blue Screen of Death (the dreaded **BSOD**). Session logging was introduced in SunPCi 2.3. SunPCi session logging provides a way for problems and failures to be captured that aid in our never-ending quest for software quality. This paper will describe the logging facility and detail some of the logging/debugging techniques that can be employed by any user to help the SunPCi product quality.

What is the SunPCi Log?

The SunPCi log file is an ASCII text file stored on the local Solaris file system. It contains informational messages, warning messages, debug information and other messages that are generated from the Solaris SunPCi drivers and shared libraries or from the Windows binaries. Many of these messages are informational, detailing such things as the Solaris or Windows environment, the revisions of the Solaris and Windows drivers, or what options have been enabled or disabled. Additionally, in the cases of detectable/recoverable errors, the log file will also contain debug information that may detail the conditions in which the error had occurred. In most cases, when a BSOD occurs, the information that is displayed on the screen is also captured in this log file. However, there are some instances of a BSOD where this information cannot be completely captured by the log file, such as if there is some fatal error in the communications protocol between the Windows and the Solaris drivers. In that case, since there is a protocol exchange breakdown, there may be no way that the BSOD information can be sent to the Solaris drivers for logging.

Where is the SunPCi Log?

The SunPCi log file always resides on the local Solaris file system. The SunPCi log file location is controlled by the logging configuration file `/etc/opt/SUNWspci2/log.conf` (Figure 1 SunPCi Log Configuration File). The configuration file contains the following entries:

- A flag disabling logging. By default, logging is enabled
- A path for the log file. By default, that path is `/var/preserve/sunpcilog`s

```
; This file controls the SunPCi logging feature. It allows logging to
; be enabled (the default) or disabled. In the LogConfig section below,
; uncomment the Status = Disabled line to disable all SunPCi logging.
;
; The log file destination can also be changed. Uncomment the Directory
; entry below and change the directory to any directory with world read
; and write permissions. Log files will be created in the specified
; directory. The default directory is /var/preserve/sunpcilog/s/

[LogConfig]
; Status = Disabled
; Directory = /var/preserve/sunpcilog/s/
```

Figure 1 SunPCi Log Configuration File

The actual log file is named `CardN.log`, where **N** is the identifying number of the card in the Solaris platform. For single SunPCi card installations, this value is 1 (e.g., the filename is `Card1.log`). For multiple SunPCi card installations, the identifying number will range with the

number of installed SunPCi cards (e.g., *Card1.log*, *Card2.log*, *Card3.log* for a system containing 3 SunPCi cards.).

Warning: Only 2 Log File Versions are Retained Per Card

Every time that a SunPCi session is started, a new log file is created. The previous version of the log file is renamed to *CardN.log.old*. Thus, only 2 versions of the log file per card exist on the system at any time – the current version and the previous version. Thus, the user must take care if a SunPCi error has occurred and the log file has captured some diagnostic information. The log file containing the diagnostic information should be preserved by copying the file to a file with a different name.

Note that if a SunPCi session is restarted (e.g., the previous SunPCi session was not exited, but the session is restarted by rebooting Windows or by selecting the *Restart* option in the SunPCi *File* menu pulldown), then the current open log file remains open and the logging continues in that log file.

What's in the SunPCi Log?

As mentioned above, the SunPCi log file contains informational and debug messages issued from the Solaris and Windows drivers. Each log file line (or “entry”) is preceded by a time stamp of when this message was logged (with the exception of the log file header – see below)..

However, it should be noted that the messages do not usually follow any standard form. In many cases, the SunPCi development team has tried to keep these messages in some standard format, but since these messages are informational and serve mainly as a debugging aid, that is not always possible.

IMPORTANT NOTE : Most, if not all messages are intended only as debugging aids for the SunPCi development team. Thus, message structure such as format, content, and retention between releases were not design considerations and may change from release to release.

Figure 2 Example SunPCi Log File and Figure 3 Example SunPCi Log File (continued) contain an example of a typical SunPCi log file. This example is from the startup of an pre-production release of SunPCi running Windows XP. A few notes about the example:

- Each line of the example logfile has been prefixed by a line number for the purposes of line identification for this document. These line numbers do not appear in the actual log file.
- This example log file was from an internal release of SunPCi (pre-2.3.1 release). Thus, some lines may not appear in log files generated in production releases, or there may exist more lines and information than appears in this example. In fact, not all log files generated from the same installation will be identical as execution of the SunPCi software (Solaris and Windows drivers) will vary from session to session.
- Not every log file entry (line) will be detailed in this example. There are some entries that are self explanatory, and some others that are transitional for the pre-release of the SunPCi software and may be absent from log files in production releases.
- Some log file lines are longer than 80 characters and therefore will tend to wrap around in the figure box.

```

1: -----
2: SunPCi 32 bit Version 2.3.1 built on Apr  6 2002 at 17:37:48
3:   Log started: Sat Apr  6 17:51:18 2002
4:   Card Name: Card1
5:   OS: SunOS
6:   Host name: ekb
7:   Release: 5.8
8:   Version: Generic_108528-09
9:   Machine: sun4u
10:  Architecture: sparc
11:  Platform: SUNW,Ultra-30
12:  Maximum open files: 1024.
13: -----
14: Sat Apr  6 17:51:18 2002 : MACAddress : 08:00:20:F0:7D:10
15: Sat Apr  6 17:51:18 2002 : main: argc 4, command line = [chimera.debug -vga -p
sunpc.xp ]
16: Sat Apr  6 17:51:18 2002 : Loading config
17: Sat Apr  6 17:51:18 2002 : X Font Server support turned OFF.
18: Sat Apr  6 17:51:18 2002 : -----
19: Sat Apr  6 17:51:18 2002 : Core dispatcher is resetting
20: Sat Apr  6 17:51:18 2002 : Dispatcher constructor for libsoftload.so.1.0,
NewSoftLoadDispatcher loaded as 1 (0x0002)
21: Sat Apr  6 17:51:21 2002 : -----
22: Sat Apr  6 17:51:21 2002 : Core dispatcher is resetting
23: Sat Apr  6 17:51:21 2002 : Dispatcher constructor for libsoftload.so.1.0,
NewSoftLoadDispatcher loaded as 1 (0x0002)
24: Sat Apr  6 17:51:31 2002 : Dispatcher constructor for libvga.so.1.1,
NewVGADispatcher failed
25: Sat Apr  6 17:51:42 2002 : FlDevOpen() failed for device /dev/rdiskette
26: Sat Apr  6 17:51:42 2002 : Memory mapped drives disabled in 32 bit mode
27: Sat Apr  6 17:51:42 2002 : Memory mapped drives disabled in 32 bit mode
28: Sat Apr  6 17:51:42 2002 : Dispatcher constructor for libint13.so.1.1,
NewInt13Dispatcher loaded as 2 (0x0004)
29: Sat Apr  6 17:52:01 2002 : [Debug]/Printf string 0xff found in sunpc.xp.ini -
SpcPrintf Mask ff
30: Sat Apr  6 17:52:01 2002 : SunBridge: Windows Version 5.1 ServicePack 0 Build 2600
31: Sat Apr  6 17:52:01 2002 : SunBridge: InitBridge: FREE Bridge.Sys for WinXP built
on Mar  4 2002 13:13:26
32: Sat Apr  6 17:52:01 2002 : SunBridge: InitBridge: Enabling bridge tracing
33: Sat Apr  6 17:52:01 2002 : SunBridge: InitBridge: KeRegisterBugCheckCallback
completed.
34: Sat Apr  6 17:52:05 2002 : SunEmDisk: DriverEntry: FREE SunEmDsk.Sys for NT 2000
built on Mar 26 2002 11:53:22
35: Sat Apr  6 17:52:05 2002 : SunEmDisk: Streaming Writes Enabled
36: Sat Apr  6 17:52:05 2002 : FlDevOpen() failed for device /dev/rdiskette
37: Sat Apr  6 17:52:05 2002 : Memory mapped drives disabled in 32 bit mode
38: Sat Apr  6 17:52:05 2002 : Memory mapped drives disabled in 32 bit mode
39: Sat Apr  6 17:52:05 2002 : Dispatcher constructor for libint13.so.1.1,
NewInt13Dispatcher loaded as 2 (0x0004)
40: Sat Apr  6 17:52:25 2002 : DriverEntry: FREE SunFlppy.Sys for NT 2000 built on Mar
25 2002 16:32:16
41: Sat Apr  6 17:52:25 2002 : SunFsd: DriverEntry: FREE SunFsd.Sys for WinXP built on
Mar 25 2002 09:20:34 SunDebugLevel 2
42: Sat Apr  6 17:52:25 2002 : NewFsdNTDispatcher: NT File System Driver protocol is
valid

```

Figure 2 Example SunPCi Log File

```

43: Sat Apr 6 17:52:25 2002 : NewFsdNTDispatcher: FSDNT File Sharing is Enabled
44: Sat Apr 6 17:52:25 2002 : Dispatcher constructor for libfsdnt.so.1.3,
NewFsdNTDispatcher loaded as 3 (0x0008)
45: Sat Apr 6 17:52:25 2002 : Dispatcher constructor for liblp.so.1.1,
NewLPDispatcher loaded as 4 (0x0010)
46: Sat Apr 6 17:52:25 2002 : Dispatcher constructor for libcopypaste.so.1.2,
NewCopyPasteDispatcher loaded as 5 (0x0020)
47: Sat Apr 6 17:52:31 2002 : DriverEntry: FREE vidbcall.sys for WinXP built on Mar
27 2002 09:59:00
48: Sat Apr 6 17:52:31 2002 : Dispatcher constructor for libvideo.so.1.5,
NewVidDispatcher loaded as 6 (0x0040)
49: Sat Apr 6 17:52:31 2002 : Font server is NOT running, dpFlags = 0x0
50: Sat Apr 6 17:52:31 2002 : SunVideo: DrvEnableDriver: CHECKED SunVideo.dll for
WinXP built on Apr 3 2002 07:29:19 - iDriverVersion 30100 cb 3 iEngineVersion 30100 c
1F SunDebugLevel 10000000
51: Sat Apr 6 17:52:31 2002 : SunVideo: Employing Old Font Routines - Font Server
Disabled
52: Sat Apr 6 17:52:31 2002 : ProcGetScreenHeightWidth: the size of the screen is
height 1024 width 1280
53: Sat Apr 6 17:52:31 2002 : SunVideo: spcgetscreenheightwidth returned from sparc
width 960 iScreenHeight 1280
54: Sat Apr 6 17:52:31 2002 : SunVideo: SetCustomVideoMode: adding mode 1280.x960.
depth 8. ...
55: Sat Apr 6 17:52:31 2002 : SunVideo: SetCustomVideoMode: adding mode 1280.x960.
depth 16. ...
56: Sat Apr 6 17:52:31 2002 : SunVideo: SetCustomVideoMode: adding mode 1280.x960.
depth 24. ...
57: Sat Apr 6 17:52:31 2002 : SunVideo: Added from sparc info height 960, Width 1280
58: Sat Apr 6 17:52:31 2002 : SunVideo: InitVideoModes: added 3. custom modes
59: Sat Apr 6 17:52:31 2002 : DriverEntry: FREE vidbcall.sys for WinXP built on Mar
27 2002 09:59:00
60: Sat Apr 6 17:52:31 2002 : Font server is NOT running, dpFlags = 0x0
61: Sat Apr 6 17:52:31 2002 : SunVideo: DrvEnableDriver: CHECKED SunVideo.dll for
WinXP built on Apr 3 2002 07:29:19 - iDriverVersion 30100 cb 3 iEngineVersion 30100 c
1F SunDebugLevel 10000000
62: Sat Apr 6 17:52:31 2002 : SunVideo: Employing Old Font Routines - Font Server
Disabled
63: Sat Apr 6 17:52:31 2002 : SunVideo: SetCustomVideoMode: adding mode 1280.x960.
depth 8. ...
64: Sat Apr 6 17:52:31 2002 : SunVideo: SetCustomVideoMode: adding mode 1280.x960.
depth 16. ...
65: Sat Apr 6 17:52:31 2002 : SunVideo: SetCustomVideoMode: adding mode 1280.x960.
depth 24. ...
66: Sat Apr 6 17:52:31 2002 : SunVideo: Added from sparc info height 960, Width 1280
67: Sat Apr 6 17:52:31 2002 : SunVideo: InitVideoModes: added 3. custom modes
68: Sat Apr 6 17:52:32 2002 : SunVideo: DrvGetDirectDrawInfo: Direct Draw Enable
Check...
69: Sat Apr 6 17:52:32 2002 : SunVideo: DrvGetDirectDrawInfo: RegistryQuery
failure for DDrawDisabled - continuing...
70: Sat Apr 6 17:52:32 2002 : SunVideo: DrvGetDirectDrawInfo: exit - .ini entry
DDrawOn NOT found - Direct Draw DISABLED
71: Sat Apr 6 17:52:47 2002 : Dispatcher constructor for liblp.so.1.1,
NewLPDispatcher loaded as 4 (0x0010)
72: Sat Apr 6 17:52:47 2002 : CheckShutdownRequest: Sparc doesn't say to shutdown
73: Sat Apr 6 17:52:49 2002 : Dispatcher constructor for liblp.so.1.1,
NewLPDispatcher loaded as 4 (0x0010)
74: Sat Apr 6 17:52:50 2002 : Dispatcher constructor for liblp.so.1.1,
NewLPDispatcher loaded as 4 (0x0010)
75: Sat Apr 6 17:52:51 2002 : SunClip (fred (6f0)) : FREE sunclip.exe for WinXP built
on Mar 4 2002 13:33:33

```

Figure 3 Example SunPCi Log File (continued)

Log File Header

Lines 1-13 is the log header. It is delineated by a line of dash ('-') characters and, unlike other entries in the log file, is not timestamped. This header contains the build and release identification of the SunPCi software as well as some Solaris system information. Of particular interest is *Card Name* (line 4), particularly on a multiple card SunPCi installation.

Also note the *Maximum open files* (line 12) value. This is the maximum number of Solaris files that can be simultaneously open. This value becomes important if the user has applications which open many files at one time and these files reside on the Solaris file system (thus employing the Sun FileSystem Driver (FSD)). The default Solaris value is 1024 and, for most Windows applications, this is sufficient. However, there are a few applications (Notably **Norton Antivirus 2002**, or **NAV 2002**) which tend to open many files during its file scanning process. If this occurs and the *Maximum open files* limit is exceeded, SunPCi will post error messages stating "Too many open files". Windows should continue operating as this condition is detectable and recoverable, but the stability of the application causing this problem may be suspect. To increase this parameter, the user should refer to the Solaris Administration guide to see how this system parameter may be increased.

Windows Boot

The next set of entries (lines 14-23) are the first entries generated by the Solaris drivers. The MAC address of the booting card is displayed (line 14) as well as the argument list used for the invocation of the SunPCi software (line 15)..

At this point, the Solaris drivers will start to load as the SunPCi card is in the process of booting. When a Solaris driver is loaded, an log entry of the form "*Dispatch constructor for <library_shareable>, <library_dispatcher_name>, loaded as <dispatcher_number> (<MetaF12Mask>)*" as shown in lines 20, 24, 28, 44, 45, 46 and 48.

Note that it seems that lines 18-20 are repeated as lines 21-23. This is an expected double reset of the SunPCi card. The first time (lines 18-20) is the on-card BIOS in flash rom resetting and reloading the soft bios. The second time (lines 21-23) is the loading of the CMOS file to the SunPCi platform.

Note that at line 24 we get a failure for load of NewVGADispatcher. This is a benign informational message and is typical of debug/informational log entries. In fact, it is expected in this case because this session is using the internal SunPCi display rather than the external monitor display, thus obviating the need for this library.

Windows Start

At line 28, when the NewInt13Dispatcher has loaded, Windows XP is starting its boot process. At this point, SunPCi is executing the Windows XP operating system, reading the disk, installing and starting various Windows device drivers. It is at this point that the SunPCi software can determine what operating system is executing and which version, as indicated by the information posted at line 30. In this case we are running Windows Version 5.1 (the internal Microsoft version number for Windows XP), with Service Pack 0, build 2600 (Microsoft numbers their operating system builds for identification and 2600 is the build number for the released version of Windows XP).

As Windows XP boots, it loads various drivers, some of which are SunPCi Windows drivers. An example of this is at line 31. This is the SunBridge driver, built FREE (optimized) for Windows XP and it displays a date and time stamp of when it was built. Note that in line 28 and line 39 the same <library_shareable> is loaded twice. This is not unusual and it is a function of the particular Windows operating system that is executing.

As other drivers load, they post some informational messages about configuration (e.g., lines 37-38, line 60, lines 63-68) or device capabilities (e.g., line 36, line 62 and line 72). In addition the operating system (in this example, Windows XP) may load a driver multiple times, as evidenced by the loading of the Windows Print Provider driver which is loaded at lines 45, 71, 73, and 74.

How to Capture Additional Debug Information

When a problem with the SunPCi platform occurs, a critical piece of diagnostic information is this log file. However, if the problem is **reproducible**, there are some additional techniques that can be employed to enhance the diagnostic information and greatly aid in the debugging process.

The basic architecture of the SunPCi platform is a set of Windows drivers which communicate with the Solaris platform by a proprietary protocol. When a reproducible functional error occurs, the user can enable the SunPCi drivers to log major portions of the protocol requests. This functionality is known as **enhanced logging**. This helps the development team to pinpoint the set of operations that occur between the Windows and Solaris drivers causing the error. The following steps detail how to enable this enhanced logging.

CAUTION: When enhanced logging is enabled, this will cause a very large number of entries to be logged and can cause the log file to grow in terms of many megabytes, or multiple gigabytes. The SunPCi development team have experienced occasions where the log file has grown so large it has filled our file system and caused SunPCi to abort. This is particularly true if the graphics transactions are logged, as there are literally hundreds of transactions required to paint a single dialog box.

NOTE : If the problem is NOT reproducible, then enhanced logging is not an effective debugging strategy.

There are 2 methods for enabling enhanced logging, depending upon the SunPCi software version:

1. If the SunPCi software version is prior to 2.3.1, the user must create the environment variable DEVEXTENSIONS prior to starting the SunPCi session. This will cause all functional subsystem transactions to be logged and the size of the log file will be an issue.
2. If the SunPCi software version is 2.3.1 or beyond, the user need only set the *MetaF12Mask* as noted below.

Enhanced logging is then toggled **ON** by depressing the <Meta>+<F12> keys. Depressing the <Meta>+<F12> keys again will toggle enhanced debugging **OFF**.

When enhanced logging is toggled **ON**, transactions between the Windows and Solaris drivers are logged. Since every logging entry requires a write to the log file, SunPCi system performance necessarily degrades. However, even more critical is that this means that every graphics request (if the internal Sun monitor is used), every asynchronous Sun Virtual Network Adapter (NDIS) transaction (if virtual NDIS is installed), and every disk transaction is logged. Obviously, this can generate a tremendous number of log entries, thus expanding the size of the log file at a very rapid rate.

However, when debugging problems, one might be able to determine if the problem is related to a specific subsystem, most of which are mutually exclusive: the graphics are corrupted, the FSD is not accessing files correctly, NDIS is losing protocol, the Copy and Paste is not working properly, etc. Thus, it is not necessary to log the transactions for all drivers, but only for a single (or a set) of drivers. For SunPCi software versions including and beyond 2.3.1, this is done by specifying the *MetaF12Mask*. This hexadecimal mask is specified in the SunPCi.ini file (which is

stored at ~/pc/SunPC.ini - see the SunPCi White Paper "SunPCi .ini file") under the [Debug] section (see Figure 4 MetaF12 Bitmask in SunPC.ini).

```
[Debug]
MetaF12Mask=0x08
```

Figure 4 MetaF12 Bitmask in SunPC.ini

MetaF12Mask Specification (SunPCi Version 2.3.1)

The *MetaF12Mask* is an unsigned 32-bit value, each bit position maps to a specific Solaris driver (e.g., a <library_shareable>). This mask acts as a filter to define which Solaris drivers are to be logged when the user toggles enhanced logging on by entering <Meta>+<F12>. For example, line 44 in Figure 3 Example SunPCi Log File (continued) is the log entry for the loading of the FSD Solaris <library_shareable> named libfsdnt.so.1.3. The *MetaF12Mask* value is 0x0008. Thus, when <Meta>+<F12> is depressed, only messages related to the FSD protocol will be entered into the log file. If the *MetaF12Mask* was changed to a value of 0x0018, then the messages associated with the Print Provider (see line 71) would also be displayed.

If the value for the *MetaF12Mask* is set to 0xFFFFFFFF, then all functional driver protocol messages will be logged. This will produce a monumental amount of logged information and disk space for the log file will become an issue.

Basic Scenario for Enhanced Logging

The basic scenario for enabling enhanced logging is:

1. Exit the current SunPCi session
2. Examine the log file and determine which functional driver set should be logged and note the hexadecimal bit value for that functional driver set.
3. For SunPCi software versions prior to 2.3.1, set the environment variable DEVEXTENSIONS. This does not have to be set on systems running SunPCi version 2.3.1 and beyond.
4. For SunPCi software versions including 2.3.1 and beyond, modify the SunPC.ini file by adding the *MetaF12Mask* entry.
5. Start a new SunPCi session
6. Just prior to the last action which causes the problem, turn **ON** enhanced logging by keying <Meta>+<F12>
7. Continue actions which cause the problem
8. Once the problem has occurred, turn **OFF** enhanced logging by keying <Meta>+<F12>.

Table 1 Window/Solaris Driver Functionality/<library_shareable> details the mapping of Solaris and Windows functionality to specific Solaris <library_shareable> identifier.

Windows/Solaris Driver Functionality	<library_shareable> Identifier
BIOS	Libsoftload.so
VGA Graphics	Libvga.so
Basic (rudimentary) device access	Libint13.so
Sun File System (FSD)	Libfsdnt.so
Sun Vritual Network (NDIS)	Libndis.so
Print Provider	Liblp.so
Copy and Paste	Libcoppaste.so
Sun Internal Monitor (X Window Graphics)	Libvideo.so

Table 1 Window/Solaris Driver Functionality/<library_shareable>

There are some caveates concerning th *MetaF12Mask* bit masks that the user must be aware.

- The bits are mutually exclusive.
- The *MetaF12Mask* bitmask for each Solaris <library_shareable> identifier may change for different SunPCi configurations. It is dependent upon the such things as the version of the Windows operating system, the devices that have been installed, and the order in which devices have been installed/removed. Thus, two SunPCi configurations with the same Windows operating system may have different values for the <library_shareable> identifier *MetaF12Mask* bitmask. However, if the configuration remain constant, as in the case of reproducing a problem, the bitmak values will not change between SunPCi session restarts.

Enhanced Logging Example

For this example, Figure 5 Enhanced Logging contains the a portion of a log file generated when the <Meta>+<F12> keys have been depressed, the *MetaF12Mask* has been set to the hexadecimal value of 0x08. Lines 94-99 shows the value found in the SunPC.ini file and which <library_shareable> will be logged. Lines 100-234 show the Windows/Solaris driver transactions to handle some FSD requests. Lines 235-237 display what occurs when the <Meta>+<F12> keys have been depressed to toggle the enhanced logging OFF.

```
92: Mon Apr 8 05:47:23 2002 : SunClip (fred (6fc)) : FREE sunclip.exe
for WinXP built on Mar 4 2002 13:33:33
93: Mon Apr 8 05:47:23 2002 : Dispatcher constructor for
libcopypaste.so.1.2, NewCopyPasteDispatcher loaded as 5 (0x0020)
94: Mon Apr 8 05:47:33 2002 : KeyboardTranslateKey: <meta> key hit -
keycode 119. (0x77) state KEYDOWN
95: Mon Apr 8 05:47:33 2002 : KeyboardTranslateKey: <meta> key hit -
keycode 10. (0xa) state KEYDOWN
96: Mon Apr 8 05:47:33 2002 : KeyboardTranslateKey: <meta>-F12 hit -
DBG_TOGGLE
97: Mon Apr 8 05:47:33 2002 : [Debug]/MetaF12Mask - 0x8
98: Mon Apr 8 05:47:33 2002 : Dispatcher NewFsdNTDispatcher (3.)
SetDebugMode() called with DBG_TOGGLE
99: Mon Apr 8 05:47:33 2002 : KeyboardTranslateKey: <meta> key hit -
keycode 10. (0xa) state KEYUP
100: Mon Apr 8 05:47:42 2002 : NTFSD MatchPath request encountered
101: Mon Apr 8 05:47:42 2002 : length = 11.
102: Mon Apr 8 05:47:42 2002 : caseSensitive = 0.
103: Mon Apr 8 05:47:42 2002 : path = /home/peten
104: Mon Apr 8 05:47:42 2002 : Dispatcher status = 1
105: Mon Apr 8 05:47:42 2002 : match = /home/peten
106: Mon Apr 8 05:47:42 2002 : FSD Stat64 File/Dir request encountered
107: Mon Apr 8 05:47:42 2002 : Name Length = 12
108: Mon Apr 8 05:47:42 2002 : Name = /home/peten

....

226: Mon Apr 8 05:47:43 2002 : FSD Open NT Dir request encountered
227: Mon Apr 8 05:47:43 2002 : Length = 12
228: Mon Apr 8 05:47:43 2002 : Name = /home/peten
229: Mon Apr 8 05:47:43 2002 : FSD ReadNTDirEx64 request encountered
230: Mon Apr 8 05:47:43 2002 : dirp = 1
231: Mon Apr 8 05:47:43 2002 : cacheSize = 7112
232: Mon Apr 8 05:47:44 2002 : FSD StatVfs request encountered
233: Mon Apr 8 05:47:44 2002 : Length = 12
234: Mon Apr 8 05:47:44 2002 : Name = /home/peten
235: Mon Apr 8 05:47:50 2002 : KeyboardTranslateKey: <meta> key hit -
keycode 119. (0x77) state KEYDOWN
236: Mon Apr 8 05:47:51 2002 : KeyboardTranslateKey: <meta> key hit -
keycode 10. (0xa) state KEYDOWN
237: Mon Apr 8 05:47:51 2002 : KeyboardTranslateKey: <meta>-F12 hit -
DBG_TOGGLE
```

Figure 5 Enhanced Logging

Windows BSOD Logging

When a Windows Blue Screen Of Death (**BSOD**) occurs, Windows operating system has detected an unrecoverable condition or the operating system has committed some unrecoverable error. A BSOD is a Microsoft diagnostic display to aid the developers in determining what had occurred and (only possibly) what conditions had caused the failure and is the last message that the operating system can post.

For the Windows NT, Windows 2000 and Windows XP operating systems, the SunPCi log may be able to capture this information and store it in the log file. However, not all information that is posted in the **BSOD** may be saved in the log file, so there may be information missing in the log file which is present in the **BSOD** itself. While obviously not perfect, this is the best “automatic” capture of this information. For more complete diagnostic purposes, the user should copy down any/all information not contained in the log file and submit that information, along with the log file, when posting a problem request. Figure 6 BSOD Capture has an example of a Windows XP BSOD that had been captured by the SunPCi logging facility.

Unfortunately, BSOD logging for Win98 or WinME is not available.

```
243: Mon Apr 8 05:49:03 2002 : ***** SunPCi BSOD detected *****
244: Mon Apr 8 05:49:03 2002 : BugcheckHandler: ** BUGCHECK - Code 16
Params 0x9a4 0x0 0x0 0x0 **
```

Figure 6 BSOD Capture

Conclusions

The SunPCi logging facility provides a powerful, on-site debugging facility for the SunPCi product development team in the event that a functional problem occurs. Additionally, the user may significantly aid in the resolution of a problem by enabling some form of the enhanced logging functionality to more fully detail the Window and Solaris driver transactions which contribute to the problem.