SN-IA Diagnostics and Support Tools Overview

Preliminary Version (May 11, 2001)

Document Number 108-xxxx-PR1

CONTRIBUTORS
Written by Darrin Goss
Edited by Allison Gosbin
Production by Rhonda Kunsman
Engineering contributions by Michel Bourget, Nelson Crumbaker, Alexis Flores, Tim Gamerdinger, Jason Godfrey, Joe Ivanuakus, Jeff Keopp, Darin Oster, Lisa Steinmetz, and Jim Young.

Record of Revision

Version	Description
PR1	May 2001 Preliminary version to support SN-IA beta release. This document is distributed as an online version only. It cannot be ordered from Logistics.

Contents

Figures

Tables

About this Document

The *SN-IA Diagnostics and Support Tools Overview* provides an overview of the diagnostics and support tools available for SN-IA beta release systems.

This document includes the following information:

- Chapter 1, “Diagnostics,” provides an overview of several diagnostic packages available for SN-IA systems (scan tools, sMDK-based diagnostics, and online diagnostics).
- Chapter 2, “Support Tools,” provides an overview of the Embedded Support Partner (ESP) and `lcrash` support tools for SN-IA systems.

Note: You can find additional information about most of these topics in the SGI 3000 series documentation set. Use this document to determine the differences between the SN-IA and SGI 3000 series versions of the diagnostics and support tools.

Conventions Used in this Document

This document uses the following conventions:

Italics	Variables and document titles
`Courier`	Program names, file names, and commands
`Courier Bold`	User input

Chapter 1

Diagnostics

This chapter provides an overview of the SN-IA versions of the following diagnostic tools:

- Scan tools
- sMDK-based diagnostics
- Online diagnostics

1.1 Scan Tools

The SN-IA scan tools include the same applications as the SGI 3000 series scan tools:

- `brick_scan`
- `scantest`
- `slit`
- `scantool`

To use the scan tools on SN-IA systems, you specify test configurations that target SN-IA bricks. Refer to *Scan Tools*, publication number 108-0263-PR1, for more information.

1.2 sMDK-based Diagnostics

The SN-IA sMDK-based diagnostics are similar to the SGI 3000 series sMDK-based diagnostics. Key differences are in the following areas:

- The tests and utilities that are available
- How you install the sMDK-based diagnostics
- How the sMDK binary files are packaged
- The location of the sMDK files (`/boot/efi`)
- How you run sMDK-based tests and utilities (from `Shell>` prompt)
- The sMDK commands that are available

1.2.1 SN-IA Diagnostic Tests and Utilities

The SN-IA sMDK diagnostic suite contains all SGI 3000 series sMDK diagnostic tests and utilities except `io7bt`. (Refer to Table 1-1.)

Table 1-1 Comparison of SN-IA sMDK Diagnostics to SGI 3000 Series sMDK Diagnostics

Name	Comparison to SGI 3000 Series Version
`barr` (barrier circuitry test)	Same test sections, command-line options, and output as the SGI 3000 series version
`ndir` (directory memory test)	The SN-IA version contains additional command-line options. (More information to be provided.) Section 2 of the SN-IA version is currently disabled.
`netkill` (Gigabyte System Network test)	Same test code, command-line options, and output as the SGI 3000 series `netkiller` test
`nmem` (node memory test)	The SN-IA version contains additional command-line options. (More information to be provided.) Section 2 of the SN-IA version is currently disabled.
`rtc` (real-time clock test)	Same test sections, command-line options, and output as the SGI 3000 series version
`rtrb` (router basic test)	Same test sections, command-line options, and output as the SGI 3000 series version
`rtrdmp` (router dump utility)	Same utility sections, command-line options, and output as the SGI 3000 series version
`rtrreg` (router register read / write utility)	Same utility sections, command-line options, and output as the SGI 3000 series version
`rtrt` (router traffic test)	Same test sections, command-line options, and output as the SGI 3000 series version `rtrt` fails or hangs if there are two processors per front-side bus (FSB).
`scct` (cache coherency test)	Different test sections, command-line options, and output than the SGI 3000 series version (More information to be provided.)
`slt` (system-level stress test)	Same command-line options and output as the SGI 3000 series `System_Level_Test` test There are some new features to test SN-IA hardware. (More information to be provided.)
`topology` (display topology utility)	Same utility sections, command-line options, and output as the SGI 3000 series version

Table 1-1 (continued) Comparison of SN-IA sMDK Diagnostics to SGI 3000 Series sMDK

Name	Comparison to SGI 3000 Series Version

xbg (Xbridge test)	Same test sections, command-line options, and output as the SGI 3000 series version
	However, for SN-IA sMDK, you must enter `run iocfg` to probe and load the I/O hardware configuration structure before running the xbg test:
	`Shell> lilo -t 0 -q xbg.sdk`
	`smdk> run iocfg`
	`smdk> run xbg`

Refer to *sMDK-based Field Diagnostics*, publication number 108-0264-PR1, for more information about sMDK and the sMDK-based diagnostic tests and utilities.

1.2.2 Installing the sMDK-based Diagnostics

The sMDK-based diagnostics for SN-IA systems can be loaded as a group from an `rpm` file or as individual binary files. A CD containing all the necessary files will be released.

1.2.3 sMDK File Packaging

The SN-IA sMDK-based diagnostics are contained in separate packages that you must load individually from the `Shell>` prompt. Each package includes the sMDK application and one or more diagnostic tests or utilities:

- `barr.sdk` (sMDK application and `barr` test)
- `ndir.sdk` (sMDK application and `ndir` test)
- `netkill.sdk` (sMDK application and `netkill` test)
- `nmem.sdk` (sMDK application and `nmem` test)
- `rtc.sdk` (sMDK application and `rtc` test)
- `rtrall.sdk` (sMDK application; `barr`, `rtrb`, and `rtrt` tests; and `rtrdmp`, `rtrreg`, and `topology` utilities)
- `rtrb.sdk` (sMDK application and `rtrb` test)
- `rtrdmp.sdk` (sMDK application and `rtrdmp` utility)
- `rtrreg.sdk` (sMDK application and `rtrreg` utility)
- `rtrt.sdk` (sMDK application and `rtrt` test)
- `scct.sdk` (sMDK application and `scct` test)
- `slt.sdk` (sMDK application and `slt` test)
- `topology.sdk` (sMDK application and `topology` utility)
- `xbg.sdk` (sMDK application, `iocfg`, and `xbg` test)

Once you load a diagnostic package, you can run only the tests and utilities that the package contains. To run tests or utilities in other packages, you must reboot the system and load the appropriate package from the `Shell>` prompt.

1.2.4 Location of the sMDK Files

The binary files for the SN-IA sMDK diagnostics are located in the `/boot/efi` directory instead of the `/stand/smdk` directory like the SGI 3000 series sMDK diagnostics.

1.2.5 Running sMDK Tests

Perform the following procedure to start sMDK and run a diagnostic test:

1. Power up or reset the system.
2. At the `Shell>` prompt, enter the `lilo` command to load the diagnostic package that contains the sMDK application and the test that you want to run:

   `Shell>` **`lilo -t 0 -q <`***`diagnostic_package_name`***`>`**

   Example: **`lilo -t 0 -q nmem.sdk`**
3. Enter the `run` command to load and run the diagnostic test:

   `smdk>` **`run <`***`test`***`> [`***`test_options`***`]`**

   Example: **`run nmem -e 15`**
4. Enter the `ds` command to view test status and determine the index value for the test:

   `smdk>` **`ds`**
5. Enter the `dscr` command to determine which displays are available for the test:

   `smdk>` **`dscr <`***`diag_index`***`>`**

   Example: **`dscr 1`**
6. Enter the `dscr` command to view the RUN display:

   `smdk>` **`dscr <`***`diag_index`***`> run 1`**

   Example: **`dscr 1 run 1`**
7. If the test detects an error, enter the `dscr` command to view the screen that displays the error information:

   `smdk>` **`dscr <`***`diag_index`***`> <`***`screen`***`> <`***`page`***`>`**

   Example: **`dscr 1 run 2`**
8. Enter the `drop` command to stop the test:

   `smdk>` **`drop <`***`diag_index`***`>`**

   Example: **`drop 1`**

After running a test, you must reboot the system (to the `Shell>` prompt) to run another test test or utility. Type **<Ctrl+T>** and enter the **reset** command at the L1 or L2 prompt to do this.

Note: If you load the `rtrall.sdk` diagnostic package, you do not need to reboot the system to load other tests in the `rtrall.sdk` diagnostic package.

1.2.6 Running sMDK Utilities

Perform the following procedure to start sMDK and run a diagnostic utility:

1. Power up or reset the system.
2. At the `Shell>` prompt, enter the `lilo` command to load the diagnostic package that contains the sMDK application and the test that you want to run:

   `Shell>` **lilo -t 0 -q <***diagnostic_package_name***>**

   Example: **lilo -t 0 -q rtrall.sdk**
3. Enter the `run` command to load and run the diagnostic utility:

   `smdk>` **run <***utlity_name***> <***utility_options***>**

   Example: **run rtrdmp**

After running a utility, you must reboot the system (to the `Shell>` prompt) to run another test or utility. Type **<Ctrl+T>** and enter the **reset** command at the L1 or L2 prompt to do this.

Note: If you load the `rtrall.sdk` diagnostic package, you do not need to reboot the system to load other utilities in the `rtrall.sdk` diagnostic package.

1.2.7 sMDK Command Availability

SN-IA sMDK supports the majority of the SGI 3000 series sMDK commands, except the following commands:

- System commands
  - `tlb`
- Process commands
  - `bp`
  - `rb`
  - `dpc`
  - `dvc`
  - `w`
  - `wr`
  - `ww`
  - `rw`

The SN-IA sMDK help display does not include these unsupported commands.

1.3 Online Diagnostics

The SN-IA online diagnostics run under the Linux operating system. Because the Linux operating system is not as mature as the IRIX operating system, the SN-IA online diagnostic suite is currently a subset of the SGI 3000 series online diagnostic suite.

The SN-IA online diagnostic suite includes the following tests, which are located in `/usr/diags/bin` directory:

- `oldisk` (disk test)
- `olmem` (memory test)
- `olvst` (TCP socket test)
- `pandora` (system stress test)
- `torpedo` (floating-point unit stress test)

The online diagnostics for the Linux operating system can be loaded as a group from an `rpm` file or as individual binary files. A CD containing all the necessary files will be released.

The online diagnostics use the same command-line options and display the same output as the SGI 3000 series online diagnostics. Refer to *Online Diagnostics*, publication number 108-0286-PR1, for more information about the online diagnostics.

Chapter 2

Support Tools

This chapter provides an overview of the SN-IA versions of the following support tools:

- Embedded Support Partner (ESP)
- `lcrash`

2.1 Embedded Support Partner

SN-IA systems use ESP for the Linux operating system, which is similar to ESP for the IRIX operating system. Key differences are that ESP for the Linux operating system:

- Runs under the Linux operating system
- Adds a serial number field to customer profile
- Updates the available event classes
- Updates the available event profiles
- Removes paging functionality
- Updates the hardware inventory report format
- Updates the performance monitoring rules

2.1.1 Runs under Linux Operating System

ESP has been ported to run under the Linux operating system. All major components were ported to provide similar functionality to ESP for the IRIX operating system.

2.1.2 Adds Serial Number Field to Customer Profile

On SN-IA systems, you must manually enter the serial number as part of the customer profile data. When you enter the customer profile data, be sure to enter the serial number in the `System serial number` field. (Refer to Figure 2-1.) If you do not enter the system serial number, ESP cannot return data to SGI.

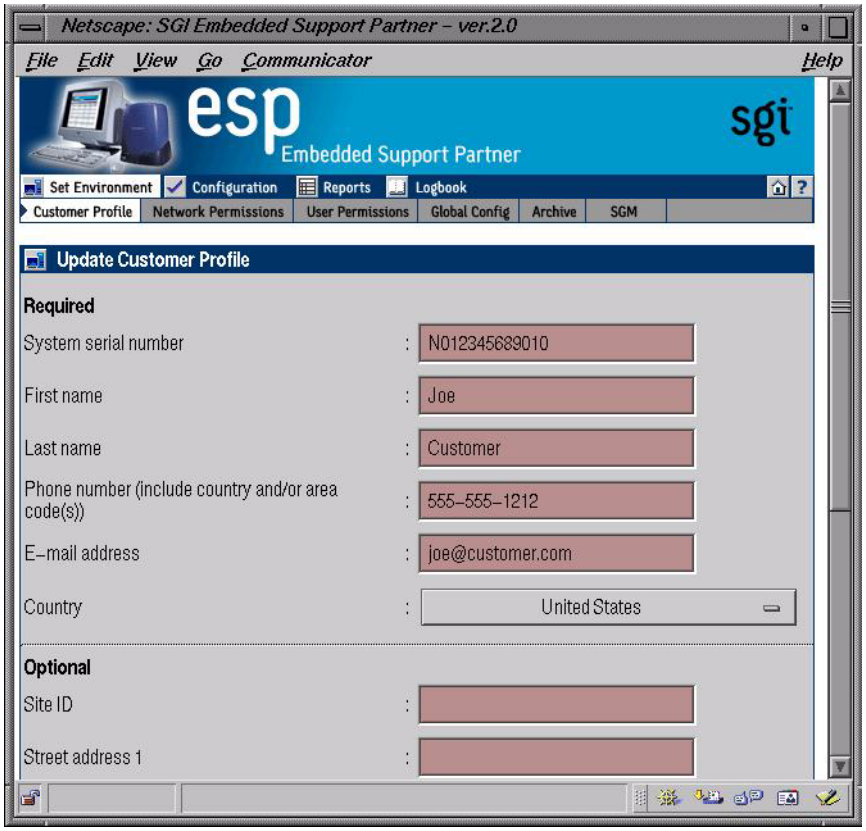

Figure 2-1 System Serial Number in Customer Profile Data

There are two ways to determine the system serial number:

- View the system inventory report.
- Use the `/usr/sbin/espsysid` command:

```
[root@euphoria /root]# espsysid
0x0000N01234568910
```

2.1.3 Updates Event Classes

ESP for the Linux operating system adds a Linux event class that contains events specific to the Linux operating system. ESP for the Linux operating system also removes event classes that are not valid for SN-IA systems.

The following event classes are available on SN-IA systems by default (refer to Figure 2-3.):

- Availability
- Diagnostic
- ESP event manager
- ESP internal events
- Linux
- System configuration

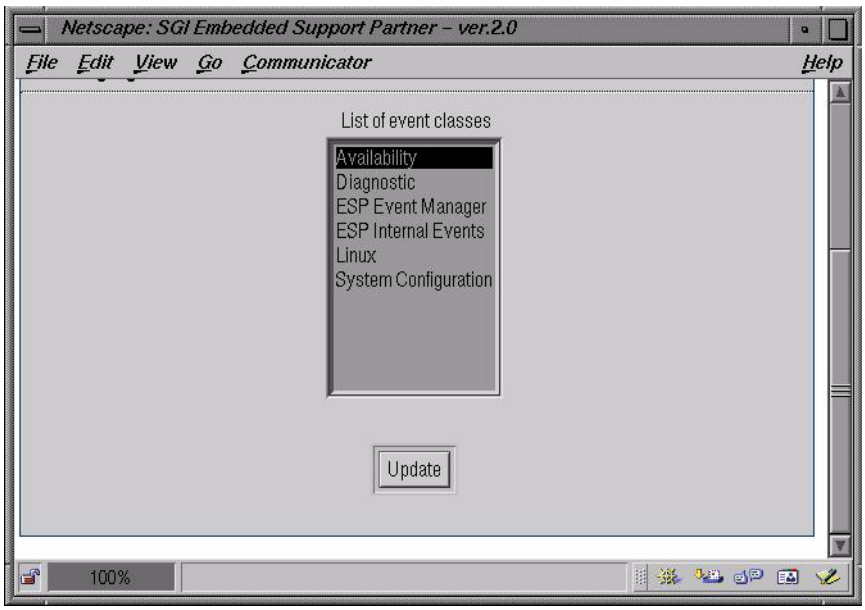

Figure 2-2 Default Event Classes

The availability, diagnostic, ESP event manager, ESP internal events, and system configuration event classes are similar to the corresponding event classes on SGI 3000 series systems.

The Linux event class contains `unix` system events (messages from `syslogd`) and kernel events (messages from `klogd`). (Figure 2-3 shows the events in the Linux event class. Table 2-1 describes the events.)

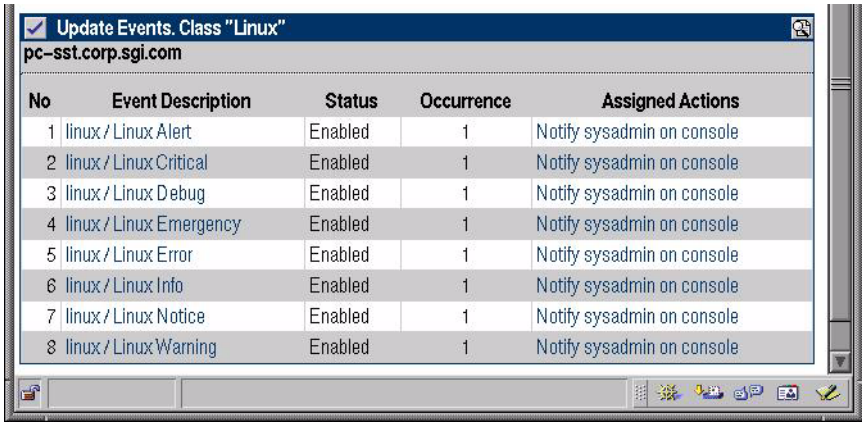

Figure 2-3 Events in the Linux Event Class

Table 2-1 Events in the Linux Event Class

Event	Description
`linux/Linux Alert`	`syslogd` and `klogd` messages from `unix` with priority `alert`
`linux/Linux Critical`	`syslogd` and `klogd` messages from `unix` with priority `crit`
`linux/Linux Debug`	`syslogd` and `klogd` messages from `unix` with priority `debug`
`linux/Linux Emergency`	`syslogd` and `klogd` messages from `unix` with priority `emerg`
`linux/Linux Error`	`syslogd` and `klogd` messages from `unix` with priority `err`
`linux/Linux Info`	`syslogd` and `klogd` messages from `unix` with priority `info`
`linux/Linux Notice`	`syslogd` and `klogd` messages from `unix` with priority `notice`
`linux/Linux Warning`	`syslogd` and `klogd` messages from `unix` with priority `warning`

2.1.4 Updates Event Profiles

ESP for the Linux operating system contains a different set of event profiles by default. (Refer to Figure 2-4.)

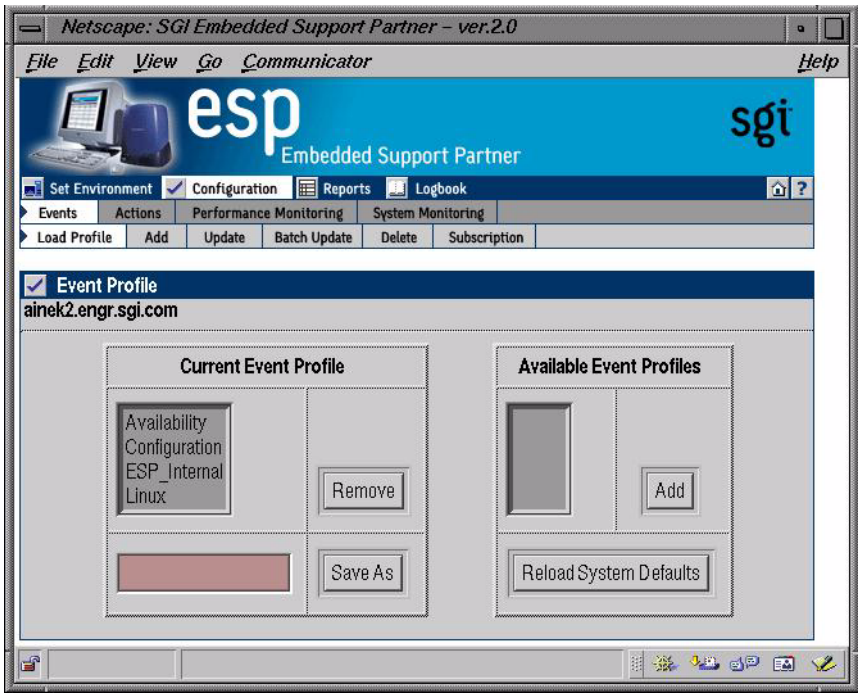

Figure 2-4 Event Profiles

The `Availability`, `Configuration`, and `ESP_Internal` event profiles are similar to the corresponding event profiles on SGI 3000 series systems. The `Linux` event profile contains configuration information for events that are specific to the Linux operating system.

2.1.5 Removes Paging Functionality

ESP for the Linux operating system does not include the paging functionality that is available in ESP for the IRIX operating system. ESP for the Linux operating system supports the following notification options:

- System console notification
- E-mail notification
- X Window System graphical user interface pop-up notification

2.1.6 Modifies the Hardware Inventory Report Format

In ESP for the Linux operating system, hardware inventory reports include detailed descriptions of each hardware component (shown in bold text). (Refer to Figure 2-5.)

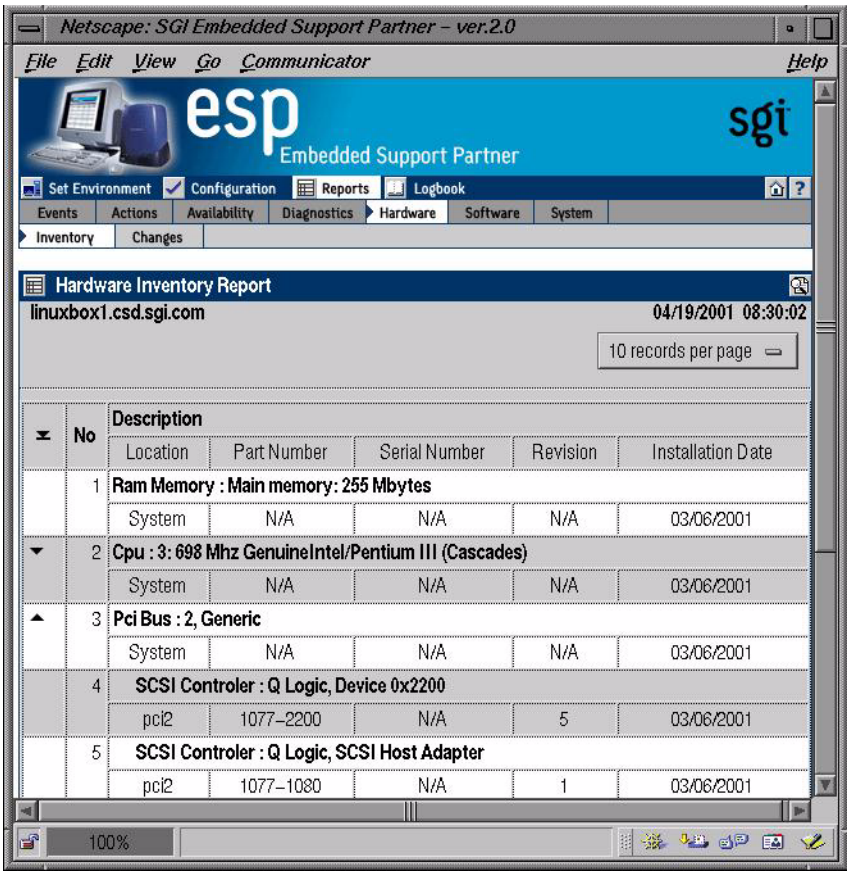

Figure 2-5 Hardware Inventory Report

2.1.7 Modifies the Performance Monitoring Rules

ESP for the Linux operating system reduces the number of performance monitoring rules that are available. (Refer to Figure 2-6.)

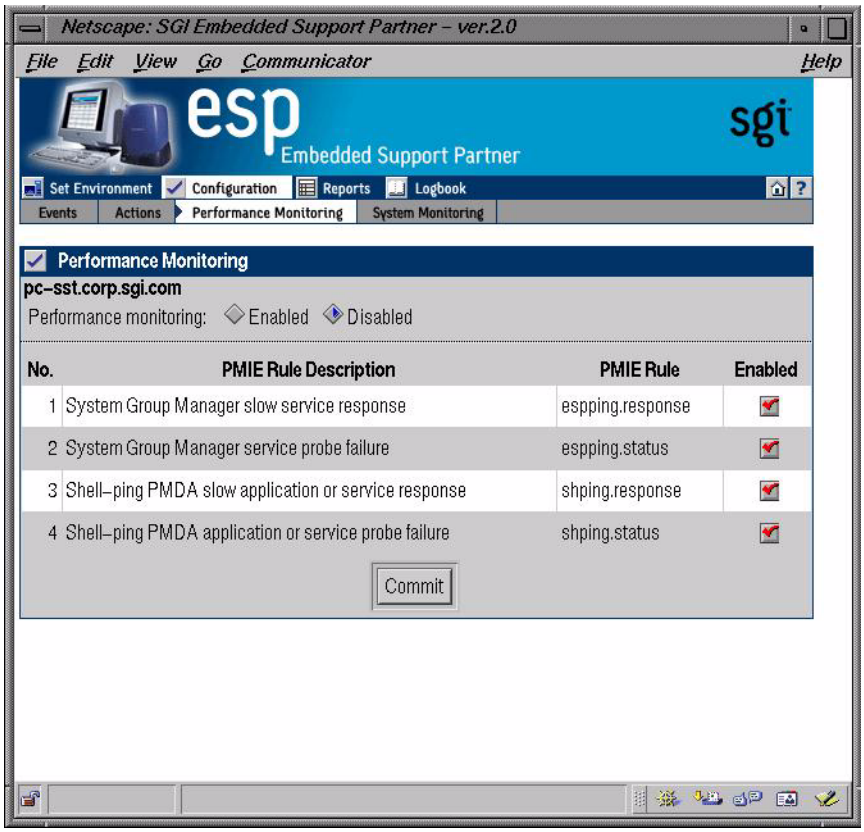

Figure 2-6 Performance Monitoring Rules

Table 2-2 describes the default performance monitoring rules that are available.

Table 2-2 Performance Monitoring Rules

Rule	Description
`espping.response`	System Group Manager slow service response: A service being monitored by group manager has taken more than threshold milliseconds to complete during the last sample interval.
`espping.status`	System Group Manager service probe failure: A service being monitored by a group manager has either failed or not responded within a timeout period during the last sample interval.
`shping.response`	Slow service response: A monitored application or service probe from the `shping` PMDA has taken more than threshold milliseconds to complete during the last sample interval.

Table 2-2 (continued) Performance Monitoring Rules

Rule	Description
shping.status	Service probe failure: An application or service being monitored by the shping PMDA has either failed or not responded within a timeout period during the last sample interval.

2.2 lcrash

lcrash is a Linux crash analysis utility; it is comparable to icrash on SGI 3000 series systems.

Use lcrash to review crash dumps (or live systems by using /dev/mem as the virtual memory and /boot/System.map as the map file) to determine why a system failure occurred. Its primary purpose is to create crash reports, but it also several built-in commands that enable you to analyze a stack trace; disassemble instructions; and review symbols, page tables, etc. It is a post-failure crash analysis tool that you can use to quickly determine why a system crash occurred.

2.2.1 lcrash FAQ Information

The following information is taken from the lcrash portion of the *Linux Kernel Crash Dumps FAQ*. The latest version of this information is available at the following URL:

http://oss.sgi.com/projects/lkcd/faq.html

What can I do with my crash dumps and lcrash?

When lcrash starts, the following output appears:

```
[root@linux]# lcrash
map = /boot/System.map, vmdump = /dev/mem, outfile = stdout, kerntypes =
/boot/Kerntypes

Please wait.................
>>
```

From the >> prompt, several commands are available to review kernel information:

```
>> ?
?               history         page            stat
addtypes        id              pb              strace
base            ldcmds          pd              sym
bt              livedump        po              symbol
deftask         md              print           t
dis             mktrace         ps              task
dt              mmap            px              trace
dump            mt              q               vtop
findsym         namelist        q!              walk
fsym            nmlist          quit            whatis
h               od              report
help            p               sizeof
```

You can use the commands to dump any portion of the vmdump memory, to review stack traces of various tasks running on the system, or to dump data structures and instructions in the kernel.

What does the stack trace output look like?

A typical stack trace looks like the following example:

```
>> t c4b38000

================================================================
                      STACK TRACE FOR TASK: 0xc4b38000 (xfs)

                       0 schedule+509 [0xc0111ce5]
                       1 schedule_timeout+103 [0xc0111a7f]
                       2 do_select+172 [0xc0130ef8]
                       3 sys_select+1009 [0xc0131479]
                       4 system_call+45 [0xc0107a65]
================================================================
```

In this example, the task enters the kernel at system_call(), and each subsequent function is a frame of the entire stack trace. In this example, system_call() calls sys_select(), which then calls do_select(), and so on. until execution reaches schedule(), where the task spins, waiting to be scheduled to run on a processor.

Each task has a different stack trace, depending on the state of the system when the kernel failure occurred. It is important to remember that the Linux operating system currently does not store processor state specific information for each processor on a system, and until this infrastructure exists, kernel panics involving multiple processors may be more difficult to debug.

What is this index.N file?

lcrash uses the index.N file against crash dumps to provide faster reads through the large vmdump.N file. The index.N file outlines where pages are located in the crash dump and whether they are compressed. The index.N file can be removed; however, it is recreated when lcrash runs again against the vmdump file. Remember to run lcrash in the directory where the crash dumps are located, so the index file is saved in the same location as the vmdump file.

What commands are available to me?

The `help` command displays all commands that are available.

The most commonly used commands are `dump`, `task`, `dis` and `trace`:

- The `dump` command displays memory contents (from either a kernel virtual or physical address).
- The `task` command displays all tasks running or scheduled to run on the system.
- The `dis` command disassembles instructions at specific memory locations.
- The `trace` command displays stack traces for various tasks.

Note: For live systems, remember to use the `deftask` command to specify the task to evaluate.

What if `lcrash` doesn't work against my crash dumps?

If `lcrash` does not work or does not appear to work correctly, it is likely that one or more of the `map`, `vmdump`, and `kerntypes` files does not match.

2.2.2 lcrash man Page

```
LCRASH(1)                                                    LCRASH(1)

NAME
       lcrash - linux crash dump analyzer

SYNOPSIS
       lcrash [ -n bounds ] map vmdump

       lcrash [ -s dumpdev dumpdir <0 | 1> ]

DESCRIPTION
       lcrash  is a hands-on utility that generates detailed ker-
       nel information in an easy-to-read  format.   lcrash  also
       provides  the  ability  to  generate  reports about system
       crash dumps.  Depending on the type of system crash  dump,
       lcrash  can  create a unique report that contains informa-
       tion about what happened when the system crashed.   lcrash
       can  be  run  on  live  systems or with any map and vmdump
       specified on the command line.

       map contains symbol table information needed for  symbolic
       access  to  the  system  memory image being examined.  The
       default map is /boot/System.map, which is used  when  ana-
       lyzing  a live system.  If the memory image being analyzed
       is from a system core dump (vmdump.N), then map must be  a
       copy  of  the  map  file that was valid at the time of the
       crash.

       vmdump is a file containing the system memory image.   The
       default  vmdump is /dev/mem, which provides access to sys-
       tem memory when analyzing a live system.  vmdump can  also
       be  a pathname to a file (vmdump.N) produced by lcrash run
       with the -s option.

OPTIONS
       -s     Allows a crash report to be  created.   The  memory
              image  is  read  from  dumpdev,  which  is the dump
              device  where  the  system  dump  memory  image  is
              placed.    The   default   dumpdev   is   /dev/vmdump,
              although it will traditionally be specified out  of
              /etc/sysconfig/vmdump,  which is where /sbin/vmdump
              reads its configuration variables from.  The report
              will be created and saved into dumpdir.  The 0 or 1
              variable declaration determines whether the  memory
              image  is  also  saved to disk along with the crash
              report.  For example,  some  system  administrators
              may  only  wish  to  create a crash report, so they
              would specify 0.  Other users may want  to  perform
              more  detailed  analysis of the system failure, and
              would specify 1.  The default value is 1.
```

-n Runs using /var/log/vmdump as the default dump directory, and tries to start lcrash using the files /var/log/vmdump/map.<bounds> and /var/log/vmdump/vmdump.<bounds>. The <bounds> variable is the number of the crash. There is a file in /var/log/vmdump called 'bounds' that has a number in it. When a crash dump is saved or reported, the 'bounds' index number is incremented so that the next index number is used for the next failure.

NOTES

1) lcrash is in a continually evolving state; it is important to get the latest lkcd and lcrash RPM/patch files to keep on top of the latest changes.

2) There are likely a number of bugs, as this is the first release. Please report any and all bugs after reviewing project details located at http://oss.sgi.com/projects/lkcd/.

Reader Comment Form

Title: SN-IA Diagnostics and Support Tools Overview **Number: 108-xxxx-PR1**

Your feedback on this publication will help us provide better documentation in the future. Please take a moment to answer the few questions below.

For what purpose did you primarily use this document?

_____Troubleshooting
_____Tutorial or introduction
_____Reference information
_____Classroom use
_____Other - please explain

_____

Using a scale from 1 (poor) to 10 (excellent), please rate this document on the following criteria and explain your ratings:

_____Accuracy _____
_____Organization _____
_____Readability _____
_____Physical qualities (binding, printing, page layout) _____
_____Amount of diagrams and photos _____
_____Quality of diagrams and photos _____

Completeness (Check one and explain your answer)

_____Too much information _____ Too little information _____Correct amount

_____

_____

_____

You may write additional comments in the space below. Mail your comments to the address below, fax them to us at +1 715 726 4353, or e-mail them to us at *spt@sgi.com.* When possible, please give specific page and paragraph references. We will respond to your comments in writing within 48 hours.

_____

_____

_____

NAME _____
JOB TITLE _____
E-MAIL ADDRESS _____
SITE/LOCATION _____
TELEPHONE _____
DATE _____

[or attach your business card]

Attn: Service Publications and Training
890 Industrial Boulevard
P.O. Box 4000
Chippewa Falls, WI 54729-0078
USA