

System Verification Program (SVP)
Reference Guide

Document Number 108-0165-002

Contributors

Written by Greg Russell
Illustrated by Greg Russell
Edited by Cindi Leiser
Production by Cindy Stief
Engineering contributions by Satish Mirle

© Copyright 1997, Silicon Graphics, Inc.— All Rights Reserved

This document contains proprietary and confidential information of Silicon Graphics, Inc. The contents of this document may not be disclosed to third parties, copied, or duplicated in any form, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

Restricted Rights Legend

Use, duplication, or disclosure of the technical data contained in this document by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013 and/or in similar or successor clauses in the FAR, or in the DOD or NASA FAR Supplement. Unpublished rights reserved under the Copyright Laws of the United States. Contractor / manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd., Mountain View, CA 94043-1389.

Silicon Graphics, Inc.
Mountain View, California

Silicon Graphics and the Silicon Graphics logo, IRIS and CHALLENGE are registered trademarks of Silicon Graphics, Inc. IRIX, Origin, Origin200, Origin2000 and InfiniteReality are trademarks of Silicon Graphics, Inc.

Contents

Introduction.....	ix
1. Introduction to the System Verification Program.....	1-1
1.1 Product Overview.....	1-1
1.2 Rationale for SVP	1-1
1.3 SVP Process.....	1-2
1.4 Current Test Coverage	1-2
1.5 List of Product Tasks and Deliverables	1-4
1.6 Description of Product Limitations	1-5
2. Getting Started.....	2-1
2.1 Installing the System Verification Program	2-1
2.1.1 Local Distribution	2-1
2.1.2 Remote Distribution	2-3
2.2 Options for Running the System Verification Program	2-3
2.3 Starting the System Verification Program	2-3
3. Viewing Test Results.....	3-1
3.1 Viewing Test Results, Configurations, and Errors.....	3-1
3.2 Analyzing Errors for Probable Causes.....	3-1
3.2.1 Memory Tests	3-2
3.2.2 Cache Tests (CHALLENGE® Systems Only)	3-3
3.2.3 Matrix Tests.....	3-3
3.2.4 Disk Tests	3-3
3.2.5 Network Tests.....	3-4
3.2.6 HIPPI and ATM Tests	3-5
4. SVP Configuration Information.....	4-1
4.1 General Description.....	4-1
4.2 Configdata.....	4-2
4.3 Patches.....	4-5
4.4 Mosaic_data	4-6

5.	The SVP Software Client	5-1
5.1	General Description.....	5-1
5.2	svp_client	5-1
5.3	e_mail_client	5-2
A.	SVP man Page	A-1
B.	Example of an SVP Report	B-1

Figures

Figure 4-1	SVP Configuration Web Page	4-7
Figure 4-2	Module Configuration Screen	4-8
Figure 4-3	Memory Configuration Screen.....	4-8
Figure 4-4	Disk Module Configuration Screen.....	4-9
Figure 4-5	Patch Information Screen.....	4-9
Figure 5-1	Message Passing System	5-1
Figure 5-2	SVP Configuration	5-3
Figure 5-3	SVP Test Results	5-3

Tables

Table 4-1	System Serial#: K0002056	4-2
Table 4-2	Memory Information	4-2
Table 4-3	Disk Information	4-2
Table 4-4	Fibre Channel RAID Information	4-3
Table 4-5	Patch Information	4-5

Introduction

This document contains information about the Silicon Graphics® service tool SVP, the System Verification Program. This information includes a general program overview; instructions on installation, configuration, and operation of this program; and descriptions of status and error reports.

The following typographic conventions are used throughout this document:

Convention	Meaning
TYPEWRITER FONT	Denotes literal items such as command names, file names, routines, directory names, path names, signals, messages, and programming language structures.
<i>italic font</i>	Denotes variable entries and words or concepts being defined.
bold typewriter font	In screen drawings of interactive sessions, denotes literal items entered by the user. Output is shown in nonbold typewriter font.
[]	Indicates an optional item.
<>	Indicates a required variable within an optional item.

Within this document, reference is made to the online man pages available under IRIX™ through the `man` command. A *man page* is a discussion of a particular element of the IRIX operating system or a compatible product.

Each man page includes a general description of one or more commands, routines, or other topics and provides details of their usage (command syntax, routine parameters, system call arguments, and so on). If more than one topic appears on a page, the entry will appear in the printed manual alphabetized only under its major name. You can access a man page named `ls` on line by typing `man ls`.

Man pages are grouped into sections numbered from 1 to 8. Each section contains entries of a particular type. Types of entries include user commands (1), administrator commands (8), system calls (2), library routines (3), file formats (5), and device descriptions (4).

Section numbers appear in parentheses after man page names. Man pages are referenced in text by entry name and section number.

Introduction to the System Verification Program

This Chapter provides an overview of the System Verification Program (SVP), explains its purpose, and lists the benefits and limitations of this service tool.

1.1 Product Overview

The System Verification Program is an IRIX™ based software tool that contains utilities and test programs, inventories the system hardware and software components, loads and runs a suite of test programs, and generates a set of files that report the results of this activity.

SVP helps a field Systems Support Engineer (SSE) to verify the configuration and functionality of a newly installed system. SVP is a system acceptance test tool, and the test result is part of the system installation sign-off requirement for the SSE.

1.2 Rationale for SVP

The SVP process is a uniform approach that is used to verify the functionality of a newly delivered system. This process will be followed for all high-end systems.

The SVP process strengthens the link between Manufacturing and the field. The testing of systems before and after shipment is a joint venture of Manufacturing and service personnel.

The SVP process ultimately reduces support costs. SVP guides the service provider to a predictable and rational conclusion of the installation process. Problems, once resolved, need not be readdressed at future installations.

SVP builds customer confidence. The SVP process assures customers that the complete shipment of hardware and software they have ordered has been delivered and that the assembled system is functional.

1.3 SVP Process

As the last step in the Origin™ system final integration-testing process, Manufacturing runs SVP to measure and document the functionality and performance of the system. This test immediately precedes the shutdown, disassembly, and packaging of the system for shipment to the customer.

The SVP results and system configuration information are automatically logged into the Mosaic database (IB Database) after the Manufacturing run. You may view this information one day after a shipment is confirmed by Manufacturing at:

http://illiad.corp.sgi.com/mosaic_queries/svp_config.html

and

http://illiad.corp.sgi.com/mosaic_queries/svp_testresult.html

You must provide your system serial number to view this information. (Refer to the subsection "SVP Client" for an explanation of the results of the information search.)

After delivery, installation, and initial power-on diagnostic testing at the customer site, the SSE retests the system with SVP, without overpack items installed. (Perform "Out of Box" tests first.) The results of this test session are compared to the results of the final Manufacturing test. Ideally, the results of both tests will match. Then the SSE should install the overpack items as the final setup of the customer configuration and run SVP again.

Subsequently, if the system installation plan requires that hardware or software modifications be made to the system, the SSE must ensure that the initial SVP session has completed successfully before the SSE proceeds with these modifications. Following system modifications, the SSE should again run SVP to establish a new configuration and return the new information to Mountain View, California (refer to the "SVP Client" subsection). Running SVP is always the final step of any hardware or software modification activity.

1.4 Current Test Coverage

Current test coverage for SVP includes:

- Peripheral confidence tests

The peripheral confidence tests check the system peripherals such as the monitor, keyboard, mouse, spaceball, CD-ROM, tape drive, tablet, video, and floppy drive. The peripheral confidence tests require that the system contains a graphics head.

Peripheral confidence tests are interactive. For each test, the SSE must intervene manually to ensure the functionality of the device. SVP tests only connected devices. Unconnected devices can be set up with the system manager from the toolchest.

- Subsystem-level diagnostics under IRIX

The board-level diagnostic set exercises the CPU, floating-point unit, memory, and I/O subsystems.

CPU and FPU functionality tests use matrix manipulations for both single-precision and double-precision floating-point numbers. Matrix tests require the FORTRAN libraries located at `/usr/lib/libftn.so`.

The memory diagnostic allocates a chunk of memory and performs bit-pattern testing. The amount of allocated memory depends on the limits set in the operating system (`memoryuse` per process). If you want to test all of memory, then increase the memory limits for each process to the maximum available. The diagnostic will determine how much memory it can allocate out of the maximum available memory (the IRIX operating system has taken a portion) and tests it.

Caution: Changing this limit parameter requires rebuilding the kernel and should be done only if you seriously doubt the integrity of memory. After you have finished testing the memory, reset the per-process-limit to the IRIX default value.

I/O subsystem diagnostics test the BASEIO, MIO, HIPPI, ATM, MSCSI, and MENET. All PCI boards will have some amount of coverage with USRPCI diagnostics. The current release limits the USRPCI diagnostics to ATM, HIPPI, and MIO. The USRPCI diagnostics perform low-level diagnosis with PIO reads/writes to check functionality of the subsystem. Some of the tests may require external cable interfaces to cover 100% functionality. Otherwise, the tests cover about 50% to 70% functionality of the board.

The Ethernet subsystem contains a test that is automatically called by SVP. This test is called the Ethernet Thrasher and is a client/server structured socket-level program that can communicate with another host system. The host system must be running the IRIX operating system. For additional information about the Ethernet Thrasher, refer to the document titled *IRIX-Based Field Diagnostics*, publication number 108-0163-001.

I/O tests exercise only mounted disks. SVP detects mounted disks and creates temporary directories from which `dd` and `tar` commands are executed. SVP then compares the test data against a source directory. Another test, the SCSI Thrasher, stresses the SCSI subsystem. This test detects all mounted disks, and using system calls, it performs parallel reads/writes. Semaphores synchronize all reads/writes on all disks. For additional information about the SCSI thrasher, refer to the document titled *IRIX-Based Field Diagnostics*, publication number 108-0163-001.

Configuration collection for the fibre channel RAID system is a new feature that has been added to SVP. Although reading and writing data on the fibre channel disks does not occur, the configuration collection process does issue pass-through commands to the disk drives, and in this way, tests the functionality of the SCSI bus for both data and commands. The configuration information that is collected is stored with the other configuration data for the system.

- IRIX command tests

The IRIX command tests check basic IRIX commands, compiler commands, network-related commands, graphics-related commands for all graphics, and commands that test OpenGL libraries. The IRIX command tests demonstrate graphics capabilities and also conduct audio or audio-related tests. All of these tests depend on the availability of supporting software. If the required software is not present, SVP skips the tests and displays messages to indicate that the software is missing.

- InfiniteReality™ Graphics diagnostic test

A prerequisite to running the InfiniteReality Graphics diagnostic test is that `irsaudit` diagnostics, bundled either in `diag.sw.IR` or in `ee.sw.gfx`, be loaded on the disk. SVP no longer executes `irsaudit` diagnostics by default. To run `irsaudit`, you must add the `-I` option when you enter the `svp` command.

Caution: If InfiniteReality graphics exists, you must run SVP from another terminal attached to the system because `irsaudit` will reset the terminal and you will lose all the results of the previous SVP tests. All demos and results of the `irsaudit` and IRIX applications tests will be displayed only on the graphics terminal.

SVP runs with the IRIX operating system, versions 5.3, 6.1, 6.2, and 6.4; automatically recognizes the system configuration; and performs appropriate tests based on this environment. (The Service Tools R&D group will create separate images of SVP for IRIX OS versions 5.3, 6.2, and 6.4. Some features of the newer images may not apply to older versions of the operating system.)

1.5 List of Product Tasks and Deliverables

SVP develops and reports the following information during its execution:

- Provides date stamp and system name
- Lists software patches installed
- Provides hardware inventory
- Provides disk inventory
- Lists mounted and unmounted file systems
- Provides software inventory
- Provides system configuration
- Performs confidence tests on peripheral devices
- Performs board-level tests
- Performs IRIX level tests
- Performs communication network tests

Refer to Appendix B to view an example of a typical SVP report listing.

1.6 Description of Product Limitations

The System Verification Program runs under the IRIX operating system. If the hardware is incapable of loading and running the operating system, you will have to troubleshoot and repair the system using other software. SVP must be run after all the system software has been installed.

When running SVP, do not run other tests or applications. If you ignore this restriction, you may receive false results during board-level tests. Problems can also arise from improper or insufficient allocation of memory or from an insufficient amount of logical swap space for diagnostics.

SVP is a test suite that calls out a subset of diagnostics and other IRIX related commands to measure the functional integrity of the system. If the system fails, you will need to use other diagnostic methods to restore the system to operation.

Getting Started

This Chapter describes how to install and operate the System Verification Program.

2.1 Installing the System Verification Program

SVP can be distributed either locally or remotely.

2.1.1 Local Distribution

Local distribution means that SVP is distributed by compact disk (CD). To install SVP on the site workstation from a CD, use the following procedure:

1. Become a superuser of the workstation that has the CD-ROM drive.

```
% /bin/su -
```

```
Password:
```

```
#
```

2. Insert the CD that contains the SVP distribution, with the label facing up, into the CD-ROM caddy and push the caddy into the CD-ROM drive.
3. Determine that the `mediad(1M)` process is running on your system by entering

```
# ps -efdl | grep mediad | grep -v grep
```

If a `mediad` process is running, skip to Step 6. Otherwise proceed to Step 4.

4. If it does not already exist, create a mount point directory called `/CDROM` on the workstation that has the CD-ROM drive by entering

```
# mkdir /CDROM
```

5. Determine whether `objectserver(1M)` is running on the workstation that has the CD-ROM drive by entering

```
# ps -efdl | grep objectserver | grep -v grep
```

If an `objectserver` process is running, run `mediad` on the workstation that has the CD-ROM drive by entering

```
# /usr/etc/mediad
```

After running `mediad`, continue to Step 6. If an `objectserver` process is not running, use the `mount(1M)` command to mount the CD.

First, run `hinv(1M)` to determine the `<cntrl>` number and the `<unit>` number on the CD-ROM drive by entering

```
# hinv | grep CDROM
    CDROM: unit 4 on SCSI controller 0
```

The `<unit>` number and the `<cntrl>` number are returned on the "CDROM" line.

Next, mount the CD on the workstation that has the CD-ROM drive. For systems running IRIX 5.3 through IRIX 6.3, enter

```
# /sbin/mount -o ro -t iso9660 /dev/scsi/sc<cntrl>d<unit>vol/CDROM
```

For systems running IRIX 6.3 (or greater), enter

```
# /sbin/mount -o ro -t iso9660 /dev/rdisk/dks<cntrl>d<unit>vol/CDROM
```

This command will properly mount the CD on the local system. If this command fails, please consult your local SGI technical support center for assistance. Otherwise, skip to Step 7.

6. Verify that the CD-ROM has mounted properly by entering

```
# ls /CDROM
```

If files appear in `/CDROM`, proceed to Step 7. If no files appear, then `mediad(1M)` must be turned off and the CD mounted by hand. To turn off `mediad`, enter

```
# /usr/etc/mediad -k
```

After turning off `mediad`, return to Step 4.

7. Once the CD is mounted, change the working directory to the path where the SVP images are located.

```
# cd <path>
```

where `<path>` is listed from Step 6.

8. Invoke `inst` by entering the command:

```
# /usr/sbin/inst -f.
```

When the `Inst` prompt appears, enter the following commands (in order shown from top to bottom):

```
Inst> list
```

```
Inst> go
```

```
Inst> quit
```

The SVP software loads from the CD to a directory, `/usr/SVP`. You are now ready to run the System Verification Program.

2.1.2 Remote Distribution

Remote installation means that SVP is available through the Internal Support Tools Web site at:

<http://ist.csd.sgi.com>

When you reach this Web page, select "Products" and then select the latest release for more information on installing software or documentation with various IRIX releases.

2.2 Options for Running the System Verification Program

The System Verification Program offers a variety of options that control the operation of the program. These options include:

- Run all tests (peripheral confidence tests, diagnostics, and IRIX commands)
- Run diagnostics only
- Run IRIX command-level tests only
- Report in verbose mode
- Set the run time
- Set the pass count

The details of running SVP with the various options are located in the SVP man page in Appendix A. The SSE selects the appropriate options according to the needs of the site.

2.3 Starting the System Verification Program

To run SVP, enter the following commands:

```
# cd /usr/SVP
# ./svp [-bCchIruvV] [-l loops] [-m duration] [-o logfile]
```

There are several new optional features:

- The `-c` option causes SVP to probe only the hardware and patch information and to update the configuration files at `/usr/SVP/CONFIG`.
- The `-I` option enables `irsaudit` testing.
- The `-V` option reports the version of SVP.

The program will ask that you enter the following information before SVP will start the test process:

- Your name
- The name of the test site

- The reason for the test run (MFG/INS/UDP/REP)
where MFG=manufacturing, INS=installation, UPD=upgrade, and REP=repair or replacement of a part. Ensure that the manufacturing mode is never selected in the field. (In manufacturing, an automated process sends data back to the IB database. In the field, this process should be controlled by the SSE.)
- The other system name (the default localhost) for network testing
- The default localhost e-mail address (default whoami), if the system has Mail setup
- Any optional one-line comment you want logged

Note: Always enter your name and the name of your workstation. If you do not know how to respond to the other requests, enter an empty field for each of them.

The System Verification Program should begin to run at this time.

Viewing Test Results

This Chapter describes how to view test results and errors, analyze errors for probable causes, and report config/results to the home office.

3.1 Viewing Test Results, Configurations, and Errors

All SVP test results are stored in the file `/usr/SVP/RESULT/svp_date.LOG`. The results of `irsaudit` are stored in the file `/usr/SVP/RESULT/irsaudit_date.LOG`. All errors are also logged into the file `/usr/SVP/RESULT/svp_date.ERROLOG`. You may view these files with your preferred editor (for example, `vi`).

All configuration information is logged into the directory `/usr/SVP/CONFIG`. This directory contains the files `configdata_1.1`, `software_installed`, `patches`, `mosaic_data.mps`, and `mosaic_data.email`. All comments that you have input during the run are inserted with a date entry into the file `/usr/SVP/CONFIG/run_reason`.

If you select the verbose option before you run SVP, the test results will indicate the actual command that is executed for each test. In both the verbose mode and the nonverbose mode, error messages provide detailed information about a failure.

IRIX command-level tests provide hints when a command cannot be run because some external software is not available. These hints indicate which software is needed to run the particular command. Generally, a test is skipped if its prerequisite software is not installed.

Refer to Appendix B for an example of a nonverbose SVP report listing.

3.2 Analyzing Errors for Probable Causes

Following an analysis of the test results, SVP attempts to assign a probable cause to the failure of a particular test.

3.2.1 Memory Tests

The following error messages may appear during memory testing:

```
main memory test (all cpu's) failed
CPU # main memory test failed
ERROR-can't allocate buffer
```

The diagnostics attempt to load 85% less 5MB of per process system memory when the overall memory size is greater than 64MB, or 75% less 5MB of per process memory when the overall memory size is less than or equal to 64MB.

Note: If any other application is running, memory cannot be allocated to this SVP test. Ensure that no other application is running during this test.

An error may also occur if the test runs out of logical swap space. Check `/usr/adm/SYSLOG` to collect additional information about memory test errors.

If multiple, identical errors appear within a single application, determine if this error is common to all CPUs for which the test is run. If so, then the `malloc` (memory allocation) test failed and a memory board is suspect. If the `tagram` test also fails and uses the same CPU as in the memory test, then the CPU is suspect. Swap CPUs and see if the failure moves to another slot.

```
ERROR - addr:AAAA exp:BBBB got CCCC
```

(where AAAA, BBBB, and CCCC are hexadecimal numbers)

This test uses the address-under-test in a Fibonacci series to test the high-order bits. After the test writes data into each memory address, the data is regenerated and compared to the memory contents. A test failure usually indicates a problem with the memory board. However, if the failures occur only with an individual CPU, the CPU itself may be the cause.

(CHALLENGE® systems only)

```
CPU # cache test failed
ERROR: data miscompare
At page XX, address AAAA data was BBBB should be CCCC
```

(where AAAA, BBBB, and CCCC are hexadecimal numbers)

This error occurs during the tag test. The test determines the total number of pages into which the allocated memory can be divided. Then the test writes four locations and their addresses in each page. The test compares data read from these pages to expected data patterns. If the failure is associated with one CPU, move that CPU to another slot and retest. If, however, multiple CPUs report a common failing address, the cause of the failure may be the memory board.

3.2.2 Cache Tests (CHALLENGE® Systems Only)

```
ERROR: CPU # cache test failed
ERROR - memaddr.Bit: Addr XXXX XXXX XXXX XXXX
```

This test runs only when the field mode option (-f) is selected. The test writes two types of data patterns to memory: a fixed pattern and a moving inversion pattern. The contents of memory are read and compared to the written data. A difference between the written data and the read-out data could indicate a cache failure. Therefore the cache is cleared and the data pattern is read again. A persistent problem may be either a cache or memory failure. Swapping the CPU can confirm whether the failure is CPU or memory related.

3.2.3 Matrix Tests

Failures of the matrix test indicate an CPU- or FPU-related problem. The matrix test runs FORTRAN programs that solve multidimensional matrices. Typically, matrix test failures indicate that the CPU board is failing.

3.2.4 Disk Tests

A SCSI thrasher test stresses the SCSI subsystem, including the MSCSI and IO6 boards. The SCSI thrasher test forks multiple processes to the disks being tested. All disk reads and writes are synchronized by semaphores. When each subsystem finishes its random-seek write phase, it waits for all other processes to complete and then begins the read phase. The read data and the write data are compared. The test uses five data patterns.

The SCSI thrasher test generates the following output messages:

```
PASS: Diskthrasher Passed All Patterns
```

The preceding message states that the test completed successfully; it did not detect any errors.

```
ERROR: Diskthrasher Thrasher Failed. Check error.log for error
```

```
ERROR: Data ERROR on device_name
```

```
    Disk mismatched at disk offset xxxx Expected data xxxx Act. xxxx
    Disk mismatched at disk offset xxxx Expected data xxxx Act. xxxx
    Disk mismatched at disk offset xxxx Expected data xxxx Act. xxxx
    Disk mismatched at disk offset xxxx Expected data xxxx Act. xxxx
    Disk mismatched at disk offset xxxx Expected data xxxx Act. xxxx
```

These error messages indicate a read/write data compare error. Note that the disk offset value is listed in the error message.

```
Data transfer on disk # by CPU # failed
```

This error appears when one of the mounted disks fails during a data transfer. The test writes two tar images of the current system directory into two temporary directories and compares them. If the two images do not compare, the error message above is issued. This error could be caused by a failure of the disk drive, disk controller, or interconnecting cables.

3.2.5 Network Tests

The Network Thrasher test, a client/server model, stresses the network hardware on the IO6 board by sending large blocks of data to a remote system and reading the data back. The test compares the data that is returned with the data that was sent.

The Ethernet Thrasher test has the following prerequisites:

- You must have the IRIX operating system booted and running on the system that you are testing.
- You must have access to a functional remote system that is running the IRIX operating system.
- You must have disk space available on the system you are testing for the creation of temporary files that this test uses. The amount of disk space required depends on the number of clients that you specify the test will fork.
- You must correctly set the `remote host` parameter for the system you are testing.
- You must provide an IP address with a fully qualified hostname to the `/etc/hosts` file for the system that you are testing. For example:

```
150.166.14.31 bootleg.csd.sgi.com bootleg
```
- You must ensure that the remote system accepts a guest account login.

This test generates the following output messages:

```
PASS: Network Thrasher passed
```

The test completed successfully; it did not detect any errors.

```
ERROR: Network Thrasher Failed. Check error log for error
```

The test failed with an error. This error occurs when there is a data integrity error after the data transfer is completed. The failing FRU is the board that contains the network interface being tested.

```
ERROR: Remote system is not responding. Cannot continue
```

The test failed with an error. This error occurs before the test starts all clients and determines if the remote system is operational. The remote system may not be responding for many reasons (such as the network is not functioning, the remote system is down, or the test is not configured correctly.)

ERROR: Could not fork off requested number of clients

The test failed with an error. This error occurs if the number of clients is greater than the number of processes that can be forked (an IRIX system call; refer to `man fork(2)`). If this message appears, you should kill the forked processes on the remote system and the test system as follows:

1. On the remote system, enter `killall server1`.
2. On the Origin2000™ system, enter `killall c_server` and then enter `killall e_client`.

3.2.6 HIPPI and ATM Tests

The LINC DMA test is called out by SVP in superuser mode. It transfers 8 Kbytes of DMA data from the host to LINC0 on the network card (ATM or HIPPI), executes PIO reads to read back the data, and compares the transmit and receive host buffers to verify that the data was transferred correctly. The test repeats for LINC DMA engine 1 and LINC1 DMA engine 0.

SVP Configuration Information

4.1 General Description

When SVP is run at the site, it builds a database on the system disk, which is a snapshot of the hardware configuration. This snapshot is different from the one that is created during the manufacturing process because overpack items are not yet installed on the system. The information gathered at the site is essential to enable SGI to evaluate service contract renewal, service calls to the TAC, and escalation calls for serious system problems.

Origin2000™ systems can provide additional detailed board-level information such as part numbers, serial numbers, and revisions, which are embedded electronically in the NICs (Number In a Can). SVP takes advantage of the availability of this data and includes it in the configuration database.

As system configurations evolve because of fault isolation or upgrades, SVP can archive the history of these changes whenever it is run. This history is stored in directory `/usr/SVP/CONFIG` in the following files:

- `configdata_1.1` for hardware information
- `patches` for patch information
- `mosaic_data.mps` and `mosaic_data.email`

These files contain hardware and test results packaged for transmission.

4.2 Configdata

The output of `configdata_1.1` is a set of tables instead of the single table provided by `configdata`. New tables have been created to provide information about node board memories, disk drives, and RAID units.

Most of the table fields are self-explanatory. The Flag ID field in the earlier version of `configdata` has been renamed `Deinstall`; it indicates the date on which a board or disk was deinstalled. Items that have never been deinstalled are marked as `xxxxxxxxxx`.

If any part listed is missing information, the corresponding table field (for example, `Part #`) will contain `NA` instead of `???????` as in the previous version of `configdata`. Table 4-1 through Table 4-4 are examples of the configuration data that are collected by SVP.

Table 4-1 System Serial#: K0002056

Mod SI#	Mod#	Slot #	SI#	Part#	Rev	Deinstall	Install	Type
K0005150	1	io5	DYJ729	030-0927-002	B	xxxxxxxxxx	1997/07/08	FIBRE-CHANNEL
K0005150	1	io6	ENZ658	030-0927-002	B	xxxxxxxxxx	1997/07/08	FIBRE-CHANNEL
K0005150	1	n1	DAR081	013-1839-001	D	xxxxxxxxxx	1997/07/08	4PIG5_MPLN
K0005150	1	n1	DBA334	030-0733-033	J	xxxxxxxxxx	1997/07/08	IP27

Table 4-2 Memory Information

Module Serial #	Part Serial #	Module	Slot	NodeID	Memory Size	Enable
K0005150	DBA334	1	n1	0	128	Y

Table 4-3 Disk Information

Module Serial #	Module Number	Part Serial #	Slot #	Rev	Disk ID	Disk Mfg	Disk Serial #	Deinstall	Install
K0002788	1	DFM450	io1	3232	1	IBM DCHS09Y	13059385RAMSG032	xxxxxxxxxx	1997/07/18
K0002788	1	DFM490	io1	3232	1	IBM DCHS04Y	68113330RAMSG032	xxxxxxxxxx	1997/07/18
K0002780	2	CJK376	io1	3232	1	IBM DCHS09Y	68022350RAMSG032	xxxxxxxxxx	1997/07/18
K0002169	3	DBX398	io1	3232	1	IBM DCHS04Y	68032034RAMSG032	xxxxxxxxxx	1997/07/18

Note: Table 4-2 and Table 4-3 are not created for Origin200™ systems.

Table 4-4 Fibre Channel RAID Information

Module Serial #	Module	Ctrl	Part Serial #	Slot #	Rev	Disk ID	Disk Mfg	Disk Serial #	Deinstall	Install
K0005150	1	2	DYJ722	io3	SG79	9	SEAGATE ST19171F	LA142176	1997/07/11	1997/07/11
K0005150	1	2	DYJ722	io3	SG79	0	SEAGATE ST19171F	LA373435	xxxxxxxxxx	1997/07/11
K0005150	1	2	DYJ722	io3	SG79	1	SEAGATE ST19171F	LA382977	xxxxxxxxxx	1997/07/11
K0005150	1	2	DYJ722	io3	SG79	2	SEAGATE ST19171F	LA209858	xxxxxxxxxx	1997/07/11
K0005150	1	2	DYJ722	io3	SG79	3	SEAGATE ST19171F	LA421777	xxxxxxxxxx	1997/07/11
K0005150	1	2	DYJ722	io3	SG79	4	SEAGATE ST19171F	LA059195	xxxxxxxxxx	1997/07/11
K0005150	1	2	DYJ722	io3	SG79	5	SEAGATE ST19171F	LA393473	xxxxxxxxxx	1997/07/11
K0005150	1	2	DYJ722	io3	SG79	6	SEAGATE ST19171F	LA407193	xxxxxxxxxx	1997/07/11
K0005150	1	2	DYJ722	io3	SG79	7	SEAGATE ST19171F	LA407935	xxxxxxxxxx	1997/07/11
K0005150	1	2	DYJ722	io3	SG79	8	SEAGATE ST19171F	LA413783	xxxxxxxxxx	1997/07/11
K0005150	1	2	DYJ722	io3	SG79	9	SEAGATE ST19171F	LA409355	xxxxxxxxxx	1997/07/11
K0005150	1	3	DYJ722	io3	SG79	7	SEAGATE ST19171F	LA406376	1997/07/11	1997/07/11
K0005150	1	3	DYJ722	io3	SG79	0	SEAGATE ST19171F	LA416044	xxxxxxxxxx	1997/07/11
K0005150	1	3	DYJ722	io3	SG79	1	SEAGATE ST19171F	LA445367	xxxxxxxxxx	1997/07/11
K0005150	1	3	DYJ722	io3	SG79	2	SEAGATE ST19171F	LA415326	xxxxxxxxxx	1997/07/11
K0005150	1	3	DYJ722	io3	SG79	3	SEAGATE ST19171F	LA447744	xxxxxxxxxx	1997/07/11
K0005150	1	3	DYJ722	io3	SG79	4	SEAGATE ST19171F	LA407192	xxxxxxxxxx	1997/07/11
K0005150	1	3	DYJ722	io3	SG79	5	SEAGATE ST19171F	LA410097	xxxxxxxxxx	1997/07/11
K0005150	1	3	DYJ722	io3	SG79	6	SEAGATE ST19171F	LA408613	xxxxxxxxxx	1997/07/11
K0005150	1	3	DYJ722	io3	SG79	7	SEAGATE ST19171F	LA416793	xxxxxxxxxx	1997/07/11
K0005150	1	3	DYJ722	io3	SG79	8	SEAGATE ST19171F	LA432117	xxxxxxxxxx	1997/07/11
K0005150	1	3	DYJ722	io3	SG79	9	SEAGATE ST19171F	LA417004	xxxxxxxxxx	1997/07/11
K0005150	1	4	ENZ658	io6	SG79	9	SEAGATE ST19171F	LA409355	1997/07/11	1997/07/11
K0005150	1	4	ENZ658	io6	SG79	0	SEAGATE ST19171F	LA378815	xxxxxxxxxx	1997/07/11
K0005150	1	4	ENZ658	io6	SG79	1	SEAGATE ST19171F	LA403398	xxxxxxxxxx	1997/07/11
K0005150	1	4	ENZ658	io6	SG79	2	SEAGATE ST19171F	LA402907	xxxxxxxxxx	1997/07/11
K0005150	1	4	ENZ658	io6	SG79	3	SEAGATE ST19171F	LA407927	xxxxxxxxxx	1997/07/11
K0005150	1	4	ENZ658	io6	SG79	4	SEAGATE ST19171F	LA404461	xxxxxxxxxx	1997/07/11
K0005150	1	4	ENZ658	io6	SG79	5	SEAGATE ST19171F	LA416592	xxxxxxxxxx	1997/07/11

Table 4-4 (continued) Fibre Channel RAID Information

Module Serial #	Module	Ctrl	Part Serial #	Slot #	Rev	Disk ID	Disk Mfg	Disk Serial #	Deinstall	Install
K0005150	1	4	ENZ658	io6	SG79	6	SEAGATE ST19171F	LA395016	xxxxxxxxxx	1997/07/11
K0005150	1	4	ENZ658	io6	SG79	7	SEAGATE ST19171F	LA407862	xxxxxxxxxx	1997/07/11
K0005150	1	4	ENZ658	io6	SG79	8	SEAGATE ST19171F	LA249785	xxxxxxxxxx	1997/07/11
K0005150	1	4	ENZ658	io6	SG79	9	SEAGATE ST19171F	LA142176	xxxxxxxxxx	1997/07/11
K0005150	1	6	DYJ729	io5	SG79	0	SEAGATE ST19171F	LA389380	xxxxxxxxxx	1997/07/11
K0005150	1	6	DYJ729	io5	SG79	1	SEAGATE ST19171F	LA405702	xxxxxxxxxx	1997/07/11
K0005150	1	6	DYJ729	io5	SG79	2	SEAGATE ST19171F	LA142912	xxxxxxxxxx	1997/07/11
K0005150	1	6	DYJ729	io5	SG79	3	SEAGATE ST19171F	LA396993	xxxxxxxxxx	1997/07/11
K0005150	1	6	DYJ729	io5	SG79	4	SEAGATE ST19171F	LA395790	xxxxxxxxxx	1997/07/11
K0005150	1	6	DYJ729	io5	SG79	5	SEAGATE ST19171F	LA211324	xxxxxxxxxx	1997/07/11
K0005150	1	6	DYJ729	io5	SG79	6	SEAGATE ST19171F	LA407599	xxxxxxxxxx	1997/07/11
K0005150	1	6	DYJ729	io5	SG79	7	SEAGATE ST19171F	LA413533	xxxxxxxxxx	1997/07/11
K0005150	1	6	DYJ729	io5	SG79	8	SEAGATE ST19171F	LA397131	xxxxxxxxxx	1997/07/11
K0005150	1	6	DYJ729	io5	SG79	9	SEAGATE ST19171F	LA406376	xxxxxxxxxx	1997/07/11

4.3 Patches

Patch Information, as in the following example, is available at `/usr/SVP/CONFIG/patches`. The Install/Remove Flag field may contain either I or R, but the Installation Date field always indicates when a patch is installed, regardless of the flag setting.

Table 4-5 Patch Information

Patch #	Install/ Remove Flag	Installation Date	Description
SG0001809	I	07/07/97	SpeedShop 1.1 Patch
SG0001820	I	02/20/97	specfs patch for IRIX 6.4
SG0001821	I	02/20/97	fixes to hwgfs (mount, link counts, pathconf)
SG0001868	I	05/23/97	XFS patch #1 for IRIX 6.4
SG0001943	I	05/29/97	6.4 S2MP+OCTANE IP27/BASEIO proms
SG0001978	I	07/08/97	6.4 kernel rollup
SG0002061	I	05/29/97	PCI Rollup
SG0002073	I	07/07/97	SCSI tape
SG0002118	I	07/07/97	6.4 SCSI rollup
SG0002173	I	07/07/97	Hinv rollup for 6.4.1
SG0002175	I	06/27/97	fibrechanel update

4.4 Mosaic_data

During SVP execution, the `configdata` and the results are packaged into a data format that is sent to the corporate IB database. The packaged information is stored in a file named `mosaic_data.mps` for the `svp_client` program and in `mosaic_data.email` for the e-mail client at `/usr/SVP/CONFIG`. The following text illustrates a typical file:

```
#--- begin data ---
MSG_TYPE=4501
||char::Sn::K0002056::
||char::ModuleSerialNumber::K0002105::
||char::ModuleId::3::
||char::SlotNo::n1::
||char::PartSerialNumber::CEM681::
||char::PartNumber::030-0733-003::
||char::PartRev::F::
||char::CaseId:::
||char::DateRun::19961124::

MSG_TYPE=4501
||char::Sn::K0002056::
||char::ModuleSerialNumber::K0002105::
||char::ModuleId::3::
||char::SlotNo::n1::
||char::PartSerialNumber::CEM536::
||char::PartNumber::013-1547-003::
||char::PartRev::A::
||char::CaseId:::
||char::DateRun::19961124::
MSG_TYPE=4502
||char::Sn::K0002105::
||char::TestNumber::0::
||char::DateCreated::19961122::
||char::TestType::MFG::
||char::ResultFlag::P::
||char::NumberOfTest::53::
||char::Comments::Final ManufacturingTesting::
#--- end data ---
```

Each `MSG_TYPE` indicates a record in the database. The `MSG_TYPE` also defines the type of data being sent back. `MSG_TYPE 4501` is a hardware part type and `MSG_TYPE 4502` is a result type. If all tests pass, only one record is sent back; it contains the number of tests passed. If any failure is observed, the "number of tests passed" record indicates how many tests passed. There is also one additional record for each failed test, which indicates the type of test that failed.

Three new message types have been created in the most recent SVP revision: `MSG_TYPE=4503` for memory, `MSG_TYPE=4504` for disks, and `MSG_TYPE=4505` for patches. An example of each new message type follows:

```
MSG_TYPE=4503
||char::Sn::K0005150::
||char::ModuleSerialNumber::K0005150::
||char::PartSerialNumber::DBA334::
||char::SlotNo::n1::
||char::node_id::0::
```

```
||char::mem_size::128::
||char::mem_enabled::Y::
```

MSG_TYPE=4504

```
||char::Sn::K0005150::
||char::ModuleSerialNumber::K0005150::
||char::PartSerialNumber::DYJ729::
||char::SlotNo::io5::
||char::disk_rev::0200::
||char::disk_id::8::
||char::disk_manf::RAID 3::
||char::disk_serial_number::95::
||char::DateRun::19970708::
```

MSG_TYPE=4505

```
||char::Sn::K0005150::
||char::patch_number::SG0001821::
||char::description::fixes to hwgfs (mount, link counts, pathconf::
||char::DateRun::02/20/97::
```

This information is archived in the MOSAIC database where you can view it with a Web browser as in the following screen snap examples:

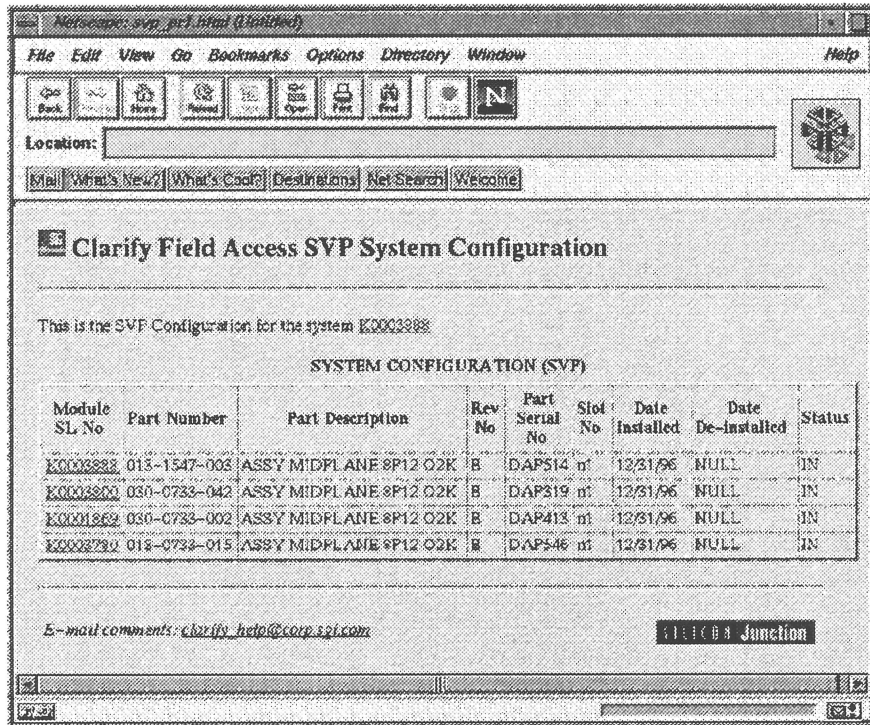


Figure 4-1 SVP Configuration Web Page

Network: svp_001.html (Default)

File Edit View Go Bookmarks Options Directory Window Help

Location: _____

What's New? What's Changed? Directories Help Search Welcome

Clarify Field Access SVP System Configuration

This is the SVP Configuration for the system: K0003888

SYSTEM CONFIGURATION (SVP)

Module SI No	Part Number	Part Description	Rev No	Part Serial No	Slot No	Date Installed	Date De-installed	Status
K0003888	013-1547-003	ASSY MIDPLANE 8P12 O2K	B	DAP514	n1	12/31/96	NULL	IN
K0003888	030-0733-003	ASSY MIDPLANE 8P12 O2K	B	DAP319	n1	12/31/96	NULL	IN
K0003888	030-0733-002	PCA IP27 GENERIC	G	DAM319	n1	12/31/96	NULL	IN
K0003888	030-0733-012	PCA IP27 GENERIC	H	DAM556	n3	12/31/96	NULL	IN
K0003888	030-0841-002	PCA ROUTER 8P O2	G	DPC570	r1	12/31/96	NULL	IN
K0003888	030-0841-002	PCA ROUTER 8P O2	G	DPC958	r2	12/31/96	NULL	IN
K0003888	030-1124-001	PCA 106 SERVER	C	DCH876	io1	12/31/96	NULL	IN

CONFIGURATION FOR MODULE K0003888 (SVP)

Part Number	Part Description	Rev No	Part Serial No	Slot No	Date Installed	Date De-installed	Status
013-1547-003	ASSY MIDPLANE 8P12 O2K	B	DAP514	n1	12/31/96	NULL	IN
030-0733-003	PCA IP27 GENERIC	G	DAM319	n2	12/31/96	NULL	IN
030-0733-003	PCA IP27 GENERIC	G	DAM415	n1	12/31/96	NULL	IN
030-0733-003	PCA IP27 GENERIC	H	DAM556	n3	12/31/96	NULL	IN
030-0841-002	PCA ROUTER 8P O2	G	DPC570	r1	12/31/96	NULL	IN
030-0841-002	PCA ROUTER 8P O2	G	DPC958	r2	12/31/96	NULL	IN
030-1124-001	PCA 106 SERVER	C	DCH876	io1	12/31/96	NULL	IN

E-mail comments: clarify.be

Figure 4-2 Module Configuration Screen

This is the SVP Configuration for the system: K0003888

CONFIGURATION FOR MODULE K0003888 (SVP)

Part Number	Part Description	Rev No	Part Serial No	Slot No	Date Installed	Date De-installed	Status
013-1547-003	ASSY MIDPLANE 8P12 O2K	B	DAP514	n1	12/31/96	NULL	IN
030-0733-003	PCA IP27 GENERIC	G	DAM319	n2	12/31/96	NULL	IN
030-0733-003	PCA IP27 GENERIC	G	DAM415	n1	12/31/96	NULL	IN
030-0733-003	PCA IP27 GENERIC	H	DAM556	n3	12/31/96	NULL	IN
030-0841-002	PCA ROUTER 8P O2	G	DPC570	r1	12/31/96	NULL	IN
030-0841-002	PCA ROUTER 8P O2	G	DPC958	r2	12/31/96	NULL	IN
030-1124-001	PCA 106 SERVER	C	DCH876	io1	12/31/96	NULL	IN

This is the SVP Configuration for the system: K0003888
Module K0003888

MEMORY CONFIGURATION FOR PART DAM319 (SVP)

Part Serial No	Slot No	Node ID	Memory Size	Enable
DAM319	n2	3	128	Y

Figure 4-3 Memory Configuration Screen

This is the SVP Configuration for the system K0003888

CONFIGURATION FOR MODULE K0003888 (SVP)

Part Number	Part Description	Rev No	Part Serial No	Slot No	Date Installed	Date De-installed	Status
013-1547-003	ASSY MIDFL ANE 9P12 O2K	B	DAP514	n1	12/31/96	NULL	IN
030-0733-003	PCA IP27 GENERIC	G	DAM319	n2	12/31/96	NULL	IN
030-0733-003	PCA IP27 GENERIC	G	DAM415	n1	12/31/96	NULL	IN
030-0733-003	PCA IP27 GENERIC	H	DAM556	n3	12/31/96	NULL	IN
030-0841-002	PCA ROUTER 8P O2	G	DPC570	r1	12/31/96	NULL	IN
030-0841-002	PCA ROUTER 8P O2	G	DPC958	r2	12/31/96	NULL	IN
030-1124-001	PCA IO6 SERVER	C	DCH876	io1	12/31/96	NULL	IN

This is the SVP Configuration for the system K0003888
Module K0003888

DISK MODULE CONFIGURATION FOR PART DCH876 (SVP)

Part Serial No	Slot No	Rev No	Disk ID	Disk Mfg	Disk Serial No	Date Installed	Date De-installed
DCH876	io1	3232	1	IBM DCHS09Y	13059385	12/31/96	NULL
DCH876	io1	3232	2	IBM DCHS04Y	68113330	12/31/96	NULL

Figure 4-4 Disk Module Configuration Screen

This is the SVP Configuration for the system K0003888

SYSTEM CONFIGURATION (SVP)

Module SL No	Part Number	Part	Patch No	Install/Remove	Description	Date Installed	Date Deinstalled
K0003888	013-1547-003	ASSY MID	SG0001691	I	rid rollopp patch #5 for 6.2, 6.3, and 6.4	02/19/97	NULL
K0003888	030-0733-042	ASSY MID	SG0001786	I	Peer SNMP Master and Encapsulator Support	04/15/97	NULL
K0001862	030-0733-003	ASSY MID	SG0001814	R	6.4 S2MP+OCTANE IP27 prom patch	02/19/97	03/01/97
K0003780	013-0733-015	ASSY MID	SG0001814	I	6.4 S2MP+OCTANE IP27 prom patch	03/05/97	NULL
			SG0001820	I	specs patch for IRIX 6.4	02/19/97	NULL
			S30001821	I	fixes to hwgfs (mount, link couns, patchconf)	02/19/97	NULL
			SG0001914	I	IRIX 6.4 libuspmda changes for PCP support	05/06/97	NULL
			SG0001868	I	XFS patch #1 for IRIX 6.4	04/23/97	NULL

E-mail comments: clarify_help@corp

Figure 4-5 Patch Information Screen

The SVP Software Client

5.1 General Description

The SVP software client is a program that returns information to the corporate Installed Base (IB) database, as a photographic view of how the system has been installed. Some customers and secure sites do not allow information to be transmitted from their sites. Consequently, this client program is not automated; it must be run as an individual command line argument after running SVP to completion. There are two mechanisms to return information to the corporate database.

5.2 `svp_client`

The first mechanism uses the MPS (Message Passing System) to transfer information to the database (refer to Figure 5-1). The mechanism is preferred because it guarantees the delivery of information to the database. If transmission failures occur, the client will indicate how many records have been successfully written to the database and then automatically update the file that is being transmitted so that it contains only unsent information. To resume the transmission of unsent information, you must reinitiate the software client.

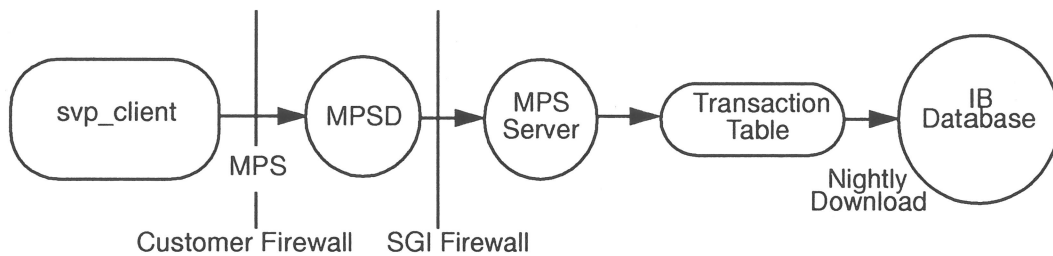


Figure 5-1 Message Passing System

The `svp_client` program consists of a shell script and a binary file. You must initiate it from the command line after the completion of SVP by entering the following two instructions:

```
cd /usr/SVP
./svp_client
```

The `svp_client` program will print out a command completion message. If any errors occur, they are reported. If only partial data is sent, a message is issued to retry the program and the datafile is automatically set to the remaining set of unsent data. After successful completion, both `mosaic_data.nips` and `mosaic_data.email` are removed.

5.3 e_mail_client

As a convenience, an e-mail client is provided in the event that the customer is not on the Internet, and the SSE must copy the `mosaic_data.email` datafile and e-mail it to SGI. Enter the following command:

```
mail svp@corp.sgi.com < /usr/SVP/CONFIG/mosaic_data.email
# rm /usr/SVP/CONFIG/mosaic_data.email
```

or

```
mail svp@corp.sgi.com < mosaic_data.email
```

The IB database is updated nightly. Data received from a site is available the following morning. A Web site has been created to view this data. It is accessible to SSEs who have access inside of the SGI firewall. The addresses for the configuration data and the test results are:

http://illiad.corp.sgi.com/mosaic_queries/svp_config.html

This is the SVP Configuration for the system K0002606

SYSTEM CONFIGURATION (SVP)								
Module SL No	Part Number	Part Description	Rev No	Part Serial No	Slot No	Date Installed	Date De-Installed	States
K0002490	013-1547-003	MIDPLANE ASSY LEG0 8P12	B	DAP223	n1	12/28/96	NULL	IN
K0002490	030-0733-003	PCA IP27 GENERIC	G	DAM318	n3	12/28/96	NULL	IN
K0002490	030-0733-003	PCA IP27 GENERIC	G	DAM402	n4	12/28/96	NULL	IN
K0002490	030-0733-003	PCA IP27 GENERIC	G	DAM871	n1	12/28/96	NULL	IN
K0002490	030-0733-003	PCA IP27 GENERIC	G	DRD109	n2	12/28/96	NULL	IN
K0002490	030-0841-002	PCA ORIGIN2000 ROUTER 8P	G	DPC493	r1	12/28/96	NULL	IN
K0002490	030-0841-002	PCA ORIGIN2000 ROUTER 8P	G	DTE801	r2	12/28/96	NULL	IN
K0002490	030-1124-001	PCA I06 SERVER	C	DFK577	io1	12/28/96	NULL	IN
K0002585	013-1547-003	MIDPLANE ASSY LEG0 8P12	B	D8F964	n1	12/28/96	NULL	IN
K0002585	030-0733-003	PCA IP27 GENERIC	G	DAM213	n1	12/28/96	NULL	IN
K0002585	030-0733-003	PCA IP27 GENERIC	G	DAM341	n4	12/28/96	NULL	IN
K0002585	030-0733-003	PCA IP27 GENERIC	G	DAM418	n3	12/28/96	NULL	IN
K0002585	030-0733-003	PCA IP27 GENERIC	G	DAM813	n2	12/28/96	NULL	IN
K0002585	030-0841-002	PCA ORIGIN2000 ROUTER 8P	G	DPC427	r1	12/28/96	NULL	IN
K0002585	030-0841-002	PCA ORIGIN2000 ROUTER 8P	G	DPC947	r2	12/28/96	NULL	IN
K0002585	030-1124-001	PCA I06 SERVER	C	DFM920	io1	12/28/96	NULL	IN
K0002606	013-1547-003	MIDPLANE ASSY LEG0 8P12	B	DAP458	n1	12/28/96	NULL	IN
K0002606	030-0733-003	PCA IP27 GENERIC	G	DAM204	n1	12/28/96	NULL	IN
K0002606	030-0733-003	PCA IP27 GENERIC	G	DAM425	n2	12/28/96	NULL	IN
K0002606	030-0733-003	PCA IP27 GENERIC	G	DAM513	n3	12/28/96	NULL	IN
K0002606	030-0733-003	PCA IP27 GENERIC	G	DAM517	n4	12/28/96	NULL	IN
K0002606	030-0841-002	PCA ORIGIN2000 ROUTER 8P	G	DLL336	r2	12/28/96	NULL	IN
K0002606	030-0841-002	PCA ORIGIN2000 ROUTER 8P	G	DTE731	r1	12/28/96	NULL	IN
K0002606	030-1124-001	PCA I06 SERVER	C	DPP047	io1	12/28/96	NULL	IN
K0002905	013-1547-003	MIDPLANE ASSY LEG0 8P12	B	DAP779	n1	12/28/96	NULL	IN
K0002905	030-0733-003	PCA IP27 GENERIC	G	DAM718	n2	12/28/96	NULL	IN
K0002905	030-0733-003	PCA IP27 GENERIC	G	DAM868	n3	12/28/96	NULL	IN
K0002905	030-0733-003	PCA IP27 GENERIC	G	DAM953	n4	12/28/96	NULL	IN
K0002905	030-0733-003	PCA IP27 GENERIC	G	DBD081	n1	12/28/96	NULL	IN
K0002905	030-0841-002	PCA ORIGIN2000 ROUTER 8P	G	DPC565	r2	12/28/96	NULL	IN
K0002905	030-0841-002	PCA ORIGIN2000 ROUTER 8P	G	DTE834	r1	12/28/96	NULL	IN
K0002905	030-1124-001	PCA I06 SERVER	C	DFK532	io1	12/28/96	NULL	IN

Figure 5-2 SVP Configuration

and

http://illiad/mosaic_queries/svp_testresult.html

This is the SVP Test Result for the system K0002606

SVP TEST RESULT					
Test No	Run Date	Test Type	Result	No of Test	Comments
0	12/28/1996	MFG	P	51	Final Configuration Test

Figure 5-3 SVP Test Results

Appendix A

SVP man Page

svp(1)

svp(1)

NAME

svp - System Verification Program

SYNOPSIS

```
svp [ -bcChIruvV ] [ -l loops ] [ -m duration ] [ -o logfile ]
```

DESCRIPTION

svp is an irix based tool that is designed to help SSE's verify the basic functionality of High End Systems during installation at the customer site. svp is a test suite that comprises peripheral confidence tests, board level diagnostics and irix command level tests. The board level diagnostics exercise CPU, FPU, Memory, I/O, Multiprocessor and graphics (KONA only). The irix command level tests cover a wide range commands, graphics, network and some application packages. Results of tests are displayed as well as saved in a logfile. A history of hardware changes get recorded whenever svp is run under /usr/SVP/CONFIG. Check in the svp users guide as to how and what should be done with the hardware configuration that is recorded. Errors are logged in svp_date.ERRORLOG under /usr/SVP/RESULT.

svp resides in /usr/SVP. To run the program you must be logged in as super-user and invoked as:

```
/usr/SVP/svp -options
```

The svp command has the options listed below:

- b disables the unix level tests from being run and executes only the board level diagnostics.
- c generates the configuration data without running any of the tests
- C disables the peripheral confidence tests. By default, the confidence tests are enabled, and are run at the beginning of the svp program. The confidence tests are interactive. The confidence tests must be exited through the exit option under file before the rest of the svp programs are run. In the event that a loop count is set for the svp program, the confidence tests are run only once.
- h displays the options that are available with the svp program.

- I option to run irsaudit diagnostics for Infinite Reality Systems
- r sets DISPLAY to current display and do not set to local host.
- u disables board level diagnostics and executes only the irix based command level tests.
- v enables verbose mode. In the verbose mode, test information is more detailed. The paths of all commands being executed is displayed. In the case of the board level diagnostics, detailed information on CPU etc is displayed. In the event of failure of any test, details of the failure are displayed on the terminal as well as logged into the logfile. By default, this option is off and only minimal information regarding the tests are displayed.
- V reports the versions number of current SVP
- l loops
option to run multiple loops of the svp program. By default, the value is set to one loop.
- m duration
option to set the runtime duration of the svp program in minutes. When set in conjunction with the -l option, the tests will stop after the runtime duration expires even in the event that the total number of loops have not been completed. By default, the -m options is disabled.
- o logfile
option to set where the results should be logged. By default, the results are logged in /usr/SVP/RESULT/svp_date.LOG.

EXAMPLE 1

```
/usr/SVP/svp -v -b -l 4 -m 120 -o SVP.log
```

The above command runs board level diagnostics only. The mode is set for verbose mode with a loop count of 4 and a maximum runtime of 120 minutes. The results are to be logged in SVP.log in the current directory.

EXAMPLE 2

```
/usr/SVP/svp
```

The above command uses all default values. The test will run peripheral confidence tests, board diagnostics and irix command level tests. Verbose mode is off and only minimal test information is provided. The loop count is set to one, with no specific time limit imposed on the test. The result will be logged in /usr/SVP/RESULT/svp_date.LOG. Errors will be logged in /usr/SVP/RESULT/svp_date.ERRORLOG.

NOTE

During the execution of the svp program, it is important that no other tests or activites be done on the installed system. This could result in incorrect results.

SEE ALSO

irsaudit(1) (infiniteReality systems only)

Appendix B

Example of an SVP Report

```
*****
*
*
*           SGI SVP Test Report           *
*
*
*****

Test start time:      Thu Jul 10 13:44:20 PDT 1997
Tester name:         sophia
Test site name:      ist
Case id:             1
System serial number: K0005150
Machine name:        perf22
Machine type:        IP27 mips
Network addr:        192.132.115.44
Host name:           perf22
Operating system:    IRIX64 6.4 Version# 02121744
svp versions:        Version 1.1
svp test options:    ./svp -C

*** Disk Inventory Information: ***

DETECT MOUNTED FILESYSTEMS
dks0dls0
dks0dls0
# c_dev.sw.c - NOT INSTALLED, some tests will be skipped
# ftn_dev.hdr.lib - NOT INSTALLED, some tests will be skipped
# c_dev.hdr.lib - NOT INSTALLED, some tests will be skipped
# ftn77_dev.sw.ftn77 - NOT INSTALLED, some tests will be skipped
# showcase.sw.showcase - NOT INSTALLED, some tests will be skipped
# x_dev.sw.dev - NOT INSTALLED, some tests will be skipped

*****
*   Board Level Tests   *
*****

*1  Test Program: Memory Tests           .....Passed
*2  Test Program: Matrix 300X300 SP      .....Passed
*3  Test Program: Matrix 300X300 DP      .....Passed
```

```

*4  Test Program: Matrix 1000X1000 SP      .....Passed
*5  Test Program: Matrix 1000X1000 DP      .....Passed
*6  Test Program: MP Matrix 1000X1000 DP   .....Passed
*7  Test Program: System Disk Test         .....Passed
Test Program: MIO Test                     .....Skipped
Test Program: SCSI Thrasher                .....Skipped

```

```

*****
*   UNIX Level Tests   *
*****

```

```

Test Program: C compiler and make         .....Skipped
Test Program: /usr/bin/strip              .....Skipped
Test Program: C compiler -n32             .....Skipped
Test Program: C compiler -64              .....Skipped
*8  Test Program: /usr/bin/size           .....Passed
Test Program: /usr/bin/nm                 .....Skipped
Test Program: /usr/bin/pixie              .....Skipped
*9  Test Program: C++ compiler            .....Passed
*10 Test Program: sn -a                    .....Passed
*11 Test Program: /bin/df -k              .....Passed
*12 Test Program: /sbin/ps                 .....Passed
*13 Test Program: /usr/bin/cat             .....Passed
*14 Test Program: /sbin/nice date          .....Passed
*15 Test Program: /usr/bin/sync            .....Passed
*16 Test Program: /usr/bsd/whereis         .....Passed
*17 Test Program: /usr/bsd/finger          .....Passed
*18 Test Program: /usr/bsd/which           .....Passed
*19 Test Program: /usr/bsd/hostid         .....Passed
*20 Test Program: /usr/bsd/last            .....Passed

```

```

*21 Test Program: /sbin/who .....Passed
Test Program: /bin/whatis .....Skipped
*22 Test Program: mpadmin -s .....Passed
*23 Test Program: /usr/sbin/ipcs .....Passed
*24 Test Program: /etc/fuser .....Passed
*25 Test Program: dircmp .....Passed
*26 Test Program: /usr/sbin/runon .....Passed
*27 Test Program: /usr/bin/time .....Passed
*28 Test Program: /usr/bin/timex .....Passed
*29 Test Program: ls -l /unix .....Passed
*30 Test Program: /bin/chmod .....Passed
*31 Test Program: /bin/chown .....Passed
*32 Test Program: /usr/bin/file .....Passed
*33 Test Program: /usr/bin/more .....Passed
*34 Test Program: man .....Passed
*35 Test Program: id .....Passed
*36 Test Program: whoami .....Passed
*37 Test Program: /bin/sort .....Passed
*38 Test Program: /bin/find .....Passed
*39 Test Program: /usr/sbin/gamma .....Passed
*40 Test Program: /bin/tail .....Passed
*41 Test Program: /usr/bin/domainname .....Passed
*42 Test Program: /usr/bin/rup .....Passed
*43 Test Program: /usr/sbin/pwck .....Passed
*44 Test Program: /sbin/tar without tape .....Passed
*45 Test Program: /sbin/cpio without tape .....Passed
*46 Test Program: /sbin/dd without tape .....Passed
*47 Test Program: /usr/sbin/bru without tape.....Passed

```

```

Test Program: Open GL/X test .....Skipped

*48 Test Program: /usr/sbin/jot .....Passed
*49 Test Program: /usr/sbin/mailbox .....Passed
*50 Test Program: /usr/bin/X11/xterm .....Passed
*51 Test Program: /usr/sbin/gr_top .....Passed
*52 Test Program: /usr/sbin/searchbook .....Passed
*53 Test Program: /usr/sbin/bongo .....Passed
*54 Test Program: /usr/sysadm(SerialDevice) .....Passed
*55 Test Program: /usr/sysadm(Disk) .....Passed
*56 Test Program: /usr/sysadm(Swap) .....Passed
*57 Test Program: /usr/sbin/swmgr .....Passed
*58 Test Program: /usr/sysadm/bin/sysmgr .....Passed
*59 Test Program: ./svp_bin_gfx_texCone .....Passed
*60 Test Program: /usr/demos/.../ideas .....Passed
Test Program: /usr/demos/.../snoop .....Skipped
*61 Test Program: /usr/demos/.../text_cube .....Passed
*62 Test Program: /usr/sbin/insight .....Passed
*63 Test Program: /usr/bin/X11/xwininfo .....Passed
*64 Test Program: /usr/sbin/xman .....Passed
*65 Test Program: /usr/bin/X11/xpsview .....Passed
*66 Test Program: /usr/sbin/imginfo .....Passed
*67 Test Program: /usr/sbin/imgview .....Passed
*68 Test Program: /usr/sbin/apanel .....Passed
*69 Test Program: /usr/demos/.../doom .....Passed
Test Program: /usr/sbin/vcp .....Skipped
*70 Test Program: /usr/etc/ping .....Passed
*71 Test Program: /usr/etc/netstat .....Passed
*72 Test Program: /usr/etc/nfsstat .....Passed
*73 Test Program: /usr/bsd/rcp 5 times .....Passed

```

*74 Test Program: mount -aERROR

Test 74 : mount -a return failed got 22 should be 0

*75 Test Program: /bin/mailPassed

*76 Test Program: /usr/sbin/MailPassed

All Automated Tests Completed

Important!! For FAILURE REPORT please check svp_err.log

check the following operations:

autoconfig

reboot

/etc/halt

/etc/init.d/network stop

/etc/init.d/network start

fsck certain file system

test vi or emacs

o*****

End Test Time: Thu Jul 10 13:55:26 PDT 1997

TEST RESULTS: Pass/75 Fail/1 Total/76

TOTAL TEST TIME: 14 minutes

