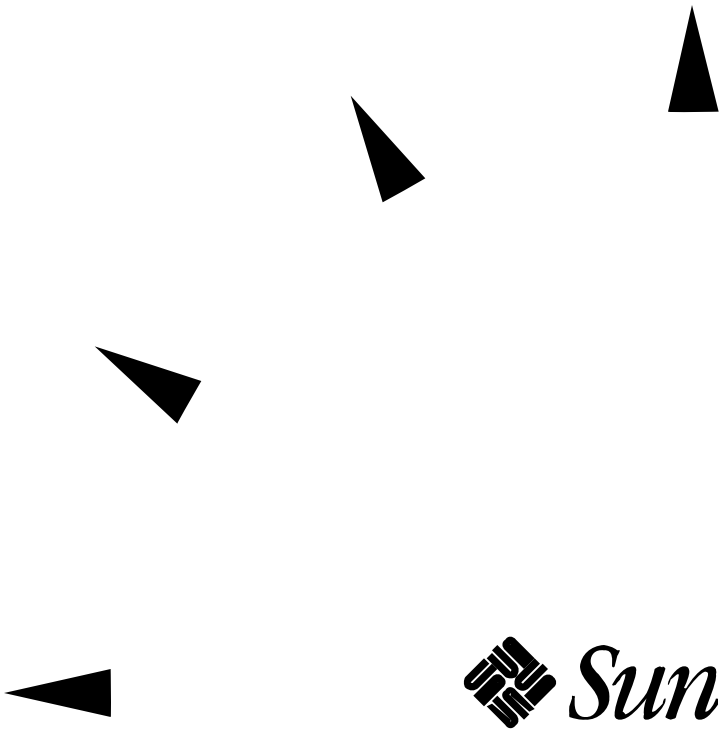
Solstice™ SyMON™ User's Guide



Sun Microsystems Computer Company
2550 Garcia Avenue
Mountain View, CA 94043 USA
415 960-1300 fax 415 969-9131

Part No: 802-5355-10
Revision A, May 1996

Sun

Please
Recycle

Contents

Tables

Figures

Preface

Solstice™ SyMON™ is a system monitoring tool that collects various system information and displays it to the user through a graphical user interface (GUI). This book, which explains how to install and use Solstice SyMON also describes how the user can modify the rules that Solstice SyMON uses to monitor the server. This allows the user to customize Solstice SyMON to meet the particular system monitoring needs of the user.

How This Book Is Organized

This manual is divided into four chapters and two appendixes:

Chapter 1, "Solstice SyMON Overview," provides an overview of the features and technical architecture of Solstice SyMON.

Chapter 2, "Installing Solstice SyMON," describes how to install and start Solstice SyMON.

Chapter 3, "Using the Consoles," provides instructions on how to navigate around the graphical user interface (GUI) of Solstice SyMON.

Chapter 4, "Understanding and Using Event Rules," describes event rules and how to modify them to meet specific monitoring needs.

Appendix A, "Kernel Reader," provides a summary of the Kernel Reader data hierarchy.

Appendix B, "Config Reader," provides a summary of the Config Reader data hierarchy.

Related Documents

Refer to the following documents for more information on the Tcl scripting language and Sun system performance tuning:

- *Tcl and the Tk Toolkit*, John K. Ousterhout, Addison-Wesley Publishing Company, 1994.
- *Practical Programming in Tcl and Tk*, by Brent B. Welch, Prentice Hall, 1995.
- Tcl Reference card; available from Specialized Systems Consultants, Inc., P.O. Box 55549, Seattle WA., 98155-0549
- *Sun Performance and Tuning*, Adrian Cockcroft, SunSoft Press, 1995.

Typographic Conventions

The following table describes the typographic conventions used in this book.

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name% su` `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called the *class* options. You *must* be root to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

Table P-2 Shell Prompts

Shell	Prompt
C shell	`machine_name%`
C shell superuser	`machine_name#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell superuser	#

Ordering Sun Documents

The SunDocs Order Desk is a distribution center for Sun Microsystems technical documents. You can use major credit cards and company purchase orders. You can order documents in the following ways:

Country	Telephone	Fax
United States	1-800-873-7869	1-800-944-0661
United Kingdom	0-800-89-88-88	0-800-89-88-87
France	05-90-61-57	05-90-61-58
Belgium	02-720-09-09	02-725-88-50
Luxembourg	32-2-720-09-09	32-2-725-88-50
Germany	01-30-81-61-91	01-30-81-61-92
The Netherlands	06-022-34-45	06-022-34-46
Sweden	020-79-57-26	020-79-57-27
Switzerland	155-19-26	155-19-27
Japan	0120-33-9096	0120-33-9097
World Wide Web: `http://www.sun.com/sunexpress/`		

Sun Welcomes Your Comments

Please use the *Reader Comment Card* that accompanies this document. We are interested in improving our documentation and welcome your comments and suggestions.

If a card is not available, you can email or fax your comments to us. Please include the part number of your document in the subject line of your email or fax message.

- Email: `smcc-docs@sun.com`
- Fax: SMCC Document Feedback
  1-415-786-6443

Solstice SyMON Overview 1

The Solstice SyMON system monitor offers powerful and simple monitoring capabilities for the Ultra™ Enterprise™ 3000, 4000, 5000, and 6000 systems using a CDE/Motif™-based Graphical User Interface (GUI), or an OPEN LOOK™ GUI.

The following sections contain this information:

- Features and benefits
- Architecture
- User Interface
- Monitoring Features

Features and Benefits

Following are the main features and benefits of Solstice SyMON:

- Identifies hardware and software conditions quickly. Conditions range from major to minor. An example of a major condition is a CPU crash and an example of a minor condition is low swap space.
- Monitors hardware performance to detect incipient hardware failure; for example, soft read errors on a disk.

To give you this critical performance information, Solstice SyMON:

- Analyzes system performance in real time; when performance problems occur, the Solstice SyMON event system alerts you if desired
- Provides graphical displays of performance data

- Maintains a log file of Solstice SyMON-detected conditions for future analysis
- Provides a graphical display of all system components and their status
- Allows the launching of the SunVTS™ diagnostic package to permit direct on-line diagnosis of hardware problems

Architecture

The Solstice SyMON internal application architecture is divided into two layers:

- Data Capture Layer (DCL)
- Management Application Program (MAP)

All monitored data originates in programs (or agents) of the DCL, which run continuously on the server. This data is sent to the MAP (which typically runs on a different machine in the network) where it is processed and displayed to the user.

The external architecture consists of the graphical user interface (GUI).

Internal Structure

The following section provides additional information on the Data Capture Layer (DCL) and Management Application Program (MAP) layers.

Data Capture Layer

The Data Capture Layer is comprised of three data-gathering agents and a daemon that controls them:

- Config Reader: This component reports system configuration data, such as the number of CPU boards, number and type of I/O boards. The Config Reader monitors and tracks changes to the configuration of the system, and the state of its components.
- Kernel Reader: This component extracts operating system data from the Solaris operating system kernel on a periodic basis. Some of the areas that are monitored and reported include:

- CPU usage
- Disk I/O
- Network I/O
- Memory usage
- Swap space usage
- Paging and swapping rates

- Log Scanner: This component monitors various system log files and searches for a user-specified list of regular expressions that are stored in the rule files. The Log Scanner generates “conditions.” The GUI can also call on the Log Scanner to directly scan the log files. The scanner is reboot-recoverable so it can capture “panic” messages immediately following a system crash.
- `symond` daemon: This component starts when the server or the event host is booted. `symond` starts the agents, monitors their activity, and restarts them if they stop. It also provides a central contact point for any clients that want to connect to the agents.

Management Application Program

The Management Application Program is comprised of the:

- Graphical User Interface (GUI): The GUI receives and displays data from all agents in the DCL.
- Event Handler: The Event Handler receives and examines incoming data from the DCL and generates event messages when appropriate. The Event Handler notifies the user of a predefined condition (as defined in the event rule file) on the system being monitored.

The Event Handler monitors changes in DCL-supplied data and compares them to the conditions established in the event rule files. When the conditions of an event are met, the Event Handler generates an event message and, depending on the action specified in the rule, may highlight the troubled component in the GUI. It also notes the occurrence of the event in its log file. When the condition for an event no longer exists, it notes the termination time of that event in the log file depending on the action specified in the rule.

The GUI displays event messages from the Event Handler.

When an event is detected, in addition to adding it to the log file and displaying it in the GUI, specific user-defined actions may also occur. User-defined actions include sending an email message, firing an SNMP trap, or launching a script.

Solstice SyMON User Interface

Solstice SyMON provides the user with a GUI to control and interact with the tool. This GUI uses a CDE/Motif-based GUI or an OPEN LOOK GUI.

Solstice SyMON Launcher and Consoles

The Launcher appears when you start Solstice SyMON. The Launcher does the following (see Figure 1-1):

- Tests for available agents (ConfigTool, Kernel Reader, Log Scanner, and the Event Handler (as in the Launcher)
- Displays the various Solstice SyMON console icons and names; each icon displays an “unavailable” symbol until it receives data from the appropriate agent
- Shows the status of agents; red indicates that the GUI is unable to contact the agent, yellow indicates that the GUI has found the agent but is not yet receiving data

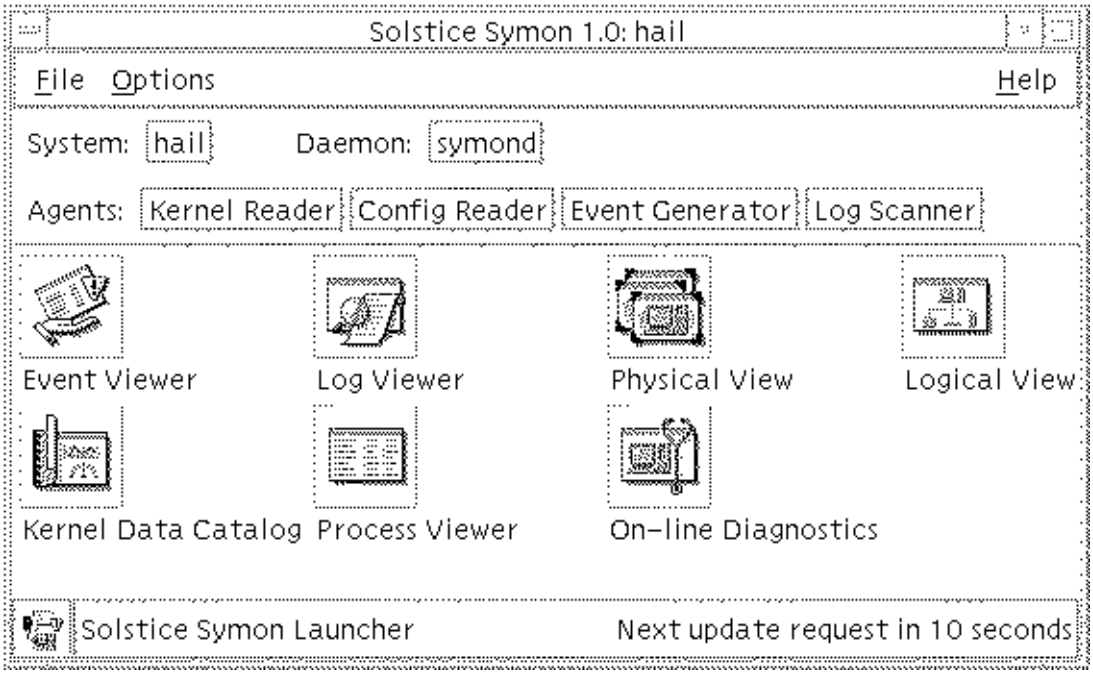


Figure 1-1 Solstice SyMON Launcher

On the Launcher, there are seven consoles. Consoles allow you to view the data collected by Solstice SyMON. Table 1-1 describes the information presented by each console.

Table 1-1 Solstice SyMON Consoles

Consoles	**Displays**
Event Viewer	Information about defined events (conditions) that occurred on the server
Log Viewer	Entries from the system log file
Physical View	Pictorial representation of the hardware of the server
Logical View	Hierarchical schematic view of the hardware components of the server

Table 1-1 Solstice SyMON Consoles *(Continued)*

Consoles	Displays
Kernel Data Catalog	Hierarchical schematic view of the monitored performance parameters
Process Viewer	Information about processes that are using system resources
On-Line Diagnostics	Connection to Sun VTS on-line diagnostics

Table 1-2 describes menu items that are typical in each console.

Table 1-2 Solstice SyMON Menu

Menu	Description
File	Gives you the option to close the window, exit the GUI, save a file or settings, or restore a file or settings
Edit	Offers different ways to manipulate windows, objects or text
View	Lists options for viewing data, or for displaying additional information about the data

Error Messages

Solstice SyMON generates error messages when a monitored operation is incorrect. Error messages are displayed in the footer of the active window.

Warning Messages

Warning messages are displayed in pop-up dialog boxes that appear above the console you are viewing. When a warning message is displayed, you must acknowledge it by selecting an action button (such as O. K. or Quit) in the dialog box.

Some warning messages state that a prerequisite for an operation has not been met. To acknowledge this message, click the Continue button before reattempting the operation.

Monitoring Features

This section provides an overview of the monitoring features of Solstice SyMON. There are three main groups of monitoring features:

- Hardware
- Software
- Alarms and events

Monitoring Hardware Status

Use Solstice SyMON to view the configuration of your system, monitor the status of system hardware components, and launch diagnostic tests. The three most frequently used consoles for monitoring hardware status are:

- Physical view
- Logical view
- On-line diagnostics

Physical View

The Physical View is a pictorial representation of the front and back of the server. The Physical View provides a picture of your system as it is actually configured. Information about each component and visual identification of a failed or troubled component is displayed.

Logical View

The Logical View is a companion view for the Physical View. While the Physical View displays the components of the system as physical entities, the Logical View displays the components in a schematic hierarchy.

On-line Diagnostics

Solstice SyMON can launch SunVTS to test and validate the CPUs, disks, and network connections of a server. Refer to the *SunVTS 2.0 User's Guide* for information about running SunVTS.

Monitoring System Software Performance

Solstice SyMON monitors software performance to isolate and identify system bottlenecks such as:

- Overloaded server
- Boards running hot

Solstice SyMON alerts the user to:

- Reconfigure software and hardware
- Plan for any system upgrades

Consoles for Monitoring Software Status

The three most frequently used consoles for monitoring software status are:

- Kernel Data Catalog
- Log Viewer
- Process Viewer

Kernel Data Catalog

The Kernel Data Catalog provides a hierarchical diagram of the system performance parameters by category (CPU, memory, disk, network, and process), much like the Logical Viewer shows a hierarchical diagram of the hardware of the system.

System Meters

The performance parameters that Solstice SyMON captures can be displayed as graphs in the System Meter console windows to show you the type and quantity of activities of the CPUs, disks, memory, and network interfaces of the server. See Table 1-3 for details.

Table 1-3 Performance Parameters

Component	Performance Parameters Captured
CPU	Performance across all CPUs on a server, or individual CPU performance; parameters that Solstice SyMON monitors are: use, system, total, wait time, idle times, context switches, interrupts, and so on
Memory	Memory available, page scan rate, total swap space, and so on
Disk	Number of active disks, projected fastest service time, number of bytes written, and average disk queue length, and so on
Network	Packet load and the number of packet errors that are occurring on the network interfaces of the server (overflow, underflow, CRC, frame, output, and input errors, as well as collisions)

Log Viewer

The Log Viewer displays text written to the log file. It searches the log using a list of regular expressions. When a match is found, the corresponding log file entry is sent to the Log View.

Process Viewer

The Process Viewer gathers and displays information about processes running on the server.

Monitoring Alarms and Events

Events are system conditions that require the attention of a system administrator. Alarms are used to alert the system administrator of an event. An alarm may be a message in the Event Viewer or a visual highlight in a console.

Event Handler

The Event Handler is the portion of Solstice SyMON that processes the performance data of the server. The Event Handler detects any event that violates the predefined performance parameters that one specified in a set of Event Rules. See Chapter 4 “Defining and Using Event Rules” for more information on the Event Handler.

The most frequently used console for monitoring events is the Event Viewer.

If the Event Handler detects problems with the system, the affected component is highlighted in red or yellow, depending on the severity of the event.

If a component is highlighted, all its related components are also highlighted. For example, if a CPU chip is highlighted, then the board on which it is mounted, and its chassis slot, are also highlighted. In addition, the Event Handler highlights the icon of the Launcher console.

Event Viewer

The Event Viewer alerts you to any problems detected by the Event Handler. Each entry displayed in the Event Viewer window represents an event and gives you the following information:

- The rule number that caused the event to be generated
- The level of severity (red, yellow, or blue)
- The time and date the event was detected
- A message indicating the type of event that was captured
- The node where the event occurred
- The priority and severity of the event (predefined, user customizable)

Installing Solstice SyMON 2

This chapter provides information on installing and initializing Solstice SyMON on an Ultra Enterprise 3000, 4000, 5000, or 6000 system. Additionally, the last part of this chapter describes how to install SNMP.

System Installation Requirements

You must install these software packages from the CD as defined in Table 2-1.

Table 2-1 Installed Software Packages

Package	Package File Name	Installed On	Disk Space
Server Subsystem	SUNWsys SUNWsyrt	Monitored server	~ 1.3 MBytes ~ < 1.0 Mbyte
Event Handler	SUNWsye SUNWsyrt	Another system	~ 1.3 MBytes ~ < 1.0 MByte
GUI	SUNWsyu SUNWsyrt	Users desktop system	~ 11.8 MBytes ~ < 1.0 MByte
man Pages	SUNWsym	Any system	~ <1.0 MByte

Follow these rules when installing the Solstice SyMON software:

- Install only one instance of the Event Handler per Server Subsystem.
- Install only one instance of the Server Subsystem per server.
- Install many instances of the GUI Subsystems per Server Subsystem.

2

Installing the Software

Determine if you are installing the software from the CD-ROM or from a remote CD-ROM, by determining if the machine on which you are doing the installation on has a CD-ROM.

There are three parts to installing Solstice SyMON:

1. Install the Server Subsystem packages.
2. Install the Event Handler packages.
3. Install the GUI Subsystem packages.

Installing From CD-ROM

Mount the CD with Volume Manager or with the `mount` command and install Solstice SyMON on the host.

▼ To Mount the CD with Volume Manager

1. **Become superuser and create the `/cdrom` directory.**
2. **Place the CD in the CD-ROM drive and close the drive door.**
   The CD is now mounted.

▼ To Mount the CD with the `mount` command

1. **Become superuser and create the `/cdrom` directory.**
2. **Place the CD in the CD-ROM drive and close the drive door.**
3. **Mount the CD-ROM by typing:**

```
# mount -r -F hsfs /dev/sr0 /cdrom
```

Installing the Server Subsystem

A general README file regarding the contents of the CD is located in the /cdrom/sup_sol_2_5_1_smcc directory. A README file for Solstice SyMON is located in the /cdrom/sup_sol_2_5_1_smcc/SMCC/Docs directory. This README file has detailed information about installing, de-installing, and activating Solstice SyMON subsystems.

Note – The SUNWsye, SUNWsyu, and SUNWsys packages depend on SUNWsyrt, the Solstice SyMON runtime library, and SUNWdtbase, the CDE-base package. If you do not install SUNWsyrt first or if you do not plan to run Solstice SyMON under the CDE environment, ignore the warning messages.

▼ To Locate the Install Directory

1. Go to the install directory:

```
# cd /cdrom/sup_sol_2_5_1_smcc
```

2. Add the packages:

```
# pkgadd -d ‘pwd‘ SUNWsyrt SUNWsys
```

Installing the Event Handler

Install the Event Handler anywhere (except on the Server Subsystem host). The following example shows how to install the Event Handler on a desktop machine.

Note – If you don't have a CD-ROM drive on the Event Handler machine, then you need to export /cdrom from another server to continue the software installation (see Steps 1, 2, and 3 below.) Otherwise, mount the CD as you did on the Server Subsystem machine.

▼ To Install the Event Handler

1. **Export the** `/cdrom` **directory from the host by performing these steps:**

   a. **Add the following line at the end of the** `/etc/dfs/dfstab` **file:**

```
share -F nfs  -o ro /cdrom
```

   b. **Start NFS and the mount daemon:**

```
# /usr/lib/nfs/nfsd -a 16
# /usr/lib/nfs/mountd
```

   c. **Export the**`/cdrom` **file system:**

```
# shareall
```

2. **Mount the** `/cdrom` **directory from the server on the desktop machine by performing these steps on the desktop machine:**

- **If the automounter is running, type**:

```
# cd /net/server_hostname/cdrom/sup_sol_2_5_1_smcc/SMCC
```

- **If the automounter is** *not* **running, type**:

```
# mkdir /cdrom
# mount server_hostname:/cdrom /cdrom
# cd /cdrom
# cd /cdrom/sup_sol_2_5_1_smcc/SMCC
```

3. **To install the Event Handler packages, type the following on the desktop machine:**

```
# pkgadd -d ‘pwd‘ SUNWsyrt SUNWsye
```

Note – If you are installing *both* the Event Handler and the GUI Subsystem on the same system, you only need to install `SUNWsyrt` once.

Installing the GUI Subsystem

You can install the GUI Subsystem on any machine. The example below shows how it is installed on a desktop machine.

Note – If you don't have a CD-ROM drive on the GUI Subsystem machine, then you need to export `/cdrom` from another server to continue the software installation (see steps 1, 2, and 3 below.) Otherwise, mount the CD as you did on the Server Subsystem machine.

▼ To Install the GUI

1. Export the `/cdrom` directory from the server.

a. On the host, add the following line at the end of the `/etc/dfs/dfstab` file:

```
# share -F nfs  -o ro -d  /cdrom/sup_sol_2_5_1_smcc
```

b. Start NFS and the mount daemon:

```
# /usr/lib/nfs/nfsd -a 16
# /usr/lib/nfs/mountd
```

c. Export the `/cdrom` file system:

```
# shareall
```

2. Mount the `/cdrom` directory from the host on the desktop machine.

- **If the automounter is running, type**:

```
# cd /net/serve hostname/cdrom/sup_sol_2_5_1_smcc/SMCC
```

- **If the automounter is *not* running, type**:

```
# mkdir /cdrom
# mount server_ hostname:/cdrom /cdrom
# cd /cdrom
# cd /cdrom/sup_sol_2_5_1_smcc/SMCC
```

3. Install the GUI Subsystem packages by typing:

```
# pkgadd -d 'pwd' SUNWsyrt SUNWsyu
```

Note – If the CDE packages are not installed, `pkgadd` will warn you that the `SUNWdtbas` and `SUNWmfrun` packages are missing. Respond `yes` to both of the warnings.

Installing the man Pages

▼ To Install the man Pages

- **If you do not have a Windex database for the man pages, become superuser and enter:**

```
# pkgadd -d 'pwd' SUNWsym
```

- **If you have a Windex database for the man pages, enter:**

```
# catman -w M /opt/SUNWsymon/man
```

Configuring Solstice SyMON

You must configure Solstice SyMON before you can run it in your environment. This process is explained in the sections that follow.

▼ To Configure the Host

♦ **Configure the host by entering:**

```
# /opt/SUNWsymon/sbin/sm_confsymon -s EventHandler_host
```

where *EventHandler_host* is the host the Event Handler is installed on.

▼ To Configure the Event Handler

♦ **Become superuser and type:**

```
# /opt/SUNWsymon/sbin/sm_confsymon -e agent_host
```

where *agent_host* is the host name of the server running the Server Subsystem.

Setting Up the User Environment

▼ To Configure the GUI

Bourne and Korn Shells

- **Add the following to** `.profile`:

```
PATH=symon_home/bin:$PATH
MANPATH=symon_home/man=$MANPATH
export PATH
export MANPATH
```

symon_home is the full path name to the installation directory, typically `/opt/SUNWsymon`.

- **If Solstice SyMON was installed in a directory other than `/opt/SUNWsymon`, add the following to `.profile`:**

```
SYMONHOME=symon_home
LD_LIBRARY_PATH=symon_home/lib:$LD_LIBRARY_PATH
export SYMONHOME
export LD_LIBRARY_PATH
```

- **If the CDE packages are not installed, add the following to `.profile`:**

```
export LD_LIBRARY_PATH
LD_LIBRARY_PATH=symon_home/lib:$LD_LIBRARY_PATH
```

- **To make the previous setup parameters effective, log out and then log in.**

C Shell

- **Add the following to `.login`:**

```
set path = ( symon_home/bin $path )
setenv MANPATH symon_home/man:$MANPATH
```

symon_home is the full path name to the installation directory, typically `/opt/SUNWsymon`.

- **If Solstice SyMON was installed in a directory other than `/opt/SUNWsymon`, add the following to `.login`:**

```
setenv SYMONHOME symon_home
setenv LD_LIBRARY_PATH symon_home/lib:$LD_LIBRARY_PATH
```

- **If the CDE packages are not installed, add the following to `.login`:**

```
setenv LD_LIBRARY_PATH symon_home/lib:$LD_LIBRARY_PATH
```

- **To make the previous setup parameters effective, type rehash at the % prompt**.

Starting Solstice SyMON

The following section describes how to start Solstice SyMON.

Starting the Server Subsystem

The Solstice SyMON Server Subsystem and the Event Handler start automatically after you reboot your system.

▼ To Start the Solstice SyMON Server Subsystem *without* rebooting

1. **Enter as the following as superuser on the Server Subsystem host:**

```
# /opt/SUNWsymon/sbin/sm_control start
```

2. **Start the Solstice SyMON Event Handler on the Event Handler machine, and enter the following as superuser:**

```
# /opt/SUNWsymon/sbin/sm_control start
```

▼ To start the Solstice SyMON Launcher

♦ **Enter the following:**

```
rehash
/opt/SUNWsymon/bin/symon -t agent_host
```

where *agent_host* is the host name of a server running the Solstice SyMON monitor agents.

Refer to the symon man page for a complete description. The symon command has a number of options and configurations.

Stopping Solstice SyMON

You can stop Solstice SyMON by following the procedures described below.

- **Enter the following as superuser to stop the Solstice SyMON Server Subsystem:**

```
# /opt/SUNWsymon/sbin/sm_control stop
```

This stops Solstice SyMON Server Subsystem. You must also stop the Solstice SyMON Event Handler on the Event Handler machine.

- **Enter the following as superuser to stop the Event Handler on the Event Handler host:**

```
# /opt/SUNWsymon/sbin/sm_control stop
```

Exiting the GUI

The Launcher window contains a File menu that includes an Exit button. After responding “yes” to the exit inquiry, the current window and all other Solstice SyMON windows are closed.

Removing the Software

Caution – `SUNWsyrt` is the runtime library. Remove `SUNWsyrt` only if all the Solstice SyMON packages are removed from that machine.

You can remove the Solstice SyMON software as described in the sections that follow. You must be superuser to remove the software.

- **To Remove the Server Subsystem, enter**

```
# pkgrm SUNWsys SUNWsyrt
```

Removing the `SUNWsys` package removes the server binaries and executables. Removing the `SUNWsyrt` package removes the runtime library.

♦ **To Remove the Event Handler, enter**

```
# pkgrm SUNWsye SUNWsyrt
```

Caution – If the GUI is also installed on this machine and you want to continue running the GUI on this machine, *do not* remove the SUNWsyrt package.

Removing the SUNWsye package removes the Event Handler binaries and data. Removing the SUNWsyrt package removes the runtime library.

♦ **To Remove the GUI, enter**

```
# pkgrm SUNWsyu SUNWsyrt
```

Removing the SUNWsyu package removes the binaries and executables. Removing the SUNWsyrt package removes the runtime library.

Caution – If the Event Handler is also installed on this machine and you want to continue running the Event Handler on this machine. *Do not* remove the SUNWsyrt package.

♦ **To Remove the man pages**

- If you do not have the Windex database of man pages, enter:

```
# pkgrm SUNWsym
```

- If you have the Windex database of man pages, enter:

```
# pkgrm SUNWsym
# rm /opt/SUNWsymon/man/windex
```

2

Installing and Setting Up SNMP Traps

The Event Handler can send SNMP traps to applications that listen for SNMP traps, such as SunNet Manager. Traps include information about events occurring on the system that may be useful to someone running a SunNet Manager console. An SNMP trap is an asynchronous message directed at a specific monitor.

To install SNMP version 2.2.3 packages on the machine named “snmphost,” refer to the SNMP installation menu.

1. **On the machine snmphost, copy the following three files in the** `/opt/SUNWsymon/etc/snm` **directory on the event host to the** `/opt/SUNWconn/agents` **directory as superuser. This is the default SNMP directory.**
   - `symon.mib.oid`
   - `symon.mib.traps`
   - `symon.mib.schema`
2. **Run the following command on the event host:**

```
# sm_confsymon -e server_host -S "list"
```

   where `-S` "*list*" indicates a space-separated list of hosts to receive the SNMP traps.

3. **Start the Event Handler daemon by entering:**

```
# /opt/SUNWsymon/sbin/cm_control start
```

4. **Run** `snm -i` **and select BasicStart.**
   Wait for the SNMP Console to come up from the main menu.
5. **Select** `Edit—>Create`**. Make sure your “Category” is** `Component` **and select your “Type” to be** `WorkStation`**; click the** `Create` **button.**

6. **A Create Form pop-up window appears; type the following:**

- Name: *snmp_hostname*
- IP address: *snmp_hostname's IP address*
- SNMP Rdcommunity: `public`
- SNMP Wrcommunity: `public`

7. **Select the** `SYMONMIB` **check button then click** `Apply`**.**

8. **Select the workstation icon you just created and from the main menu, select** `View->Event/Trap` **Report.**

9. **Type the following on the event host:**

```
/opt/SUNWsymon/sbin/trapsend -a "1.3.6.1.4.1.42 STRING (STATUS | FAIL)"
```

If you receive the message above in double quotes in your `Event/Trap` Report, you are ready to receive SNMP traps from the server.

Setting Up SNMP Hosts

Use the following command to set up SNMP hosts:

```
/opt/SUNWsymon/sbin/sm_confsymon -e server_host -S "list_of_snmphost"
```

2

Using the Consoles 3

This chapter contains instructions on how to use Solstice SyMON consoles to:

- Monitor hardware status
- Monitor system software performance
- Monitor alarms and events

This chapter also provides the basic information on available pull-down menus and associated subwindows.

After starting Solstice SyMON the Launcher appears on your screen. Chapter 4, “Solstice SyMON Overview” provides a picture of the Launcher. For information on how to start Solstice SyMON, see Chapter 4, “Installing Solstice SyMON”.

♦ **To launch a console, click on its icon in the Launcher.**

The console icons may have a red, yellow, or blue background. A colored background indicates that an event has occurred somewhere in the system that is being monitored. The colors have the following meaning:

- Red: Indicates a high severity or priority level event
- Yellow: Indicate the next highest severity or priority level event
- Blue: Indicates a capacity planning event

3

Monitoring Hardware Status

You can use Solstice SyMON to view the configuration of a system, monitor the status of its hardware components, and launch diagnostic tests.

Using the Physical View

The Physical View is a pictorial representation of the front and back of the server. Use this console for information about any system components monitored by Solstice SyMON. When you launch the Physical View, a submenu with three viewing options appears:

- Front
- Back
- Front and the back

Displaying a Detailed View of a Component

When you view a system component, you can view its internal components as well, if available. For example, when you view a CPU or memory board mounted in a chassis, you can view a more detailed image of the CPU or memory board that includes the memory and processor modules.

When a more detailed sub-view is available, the Physical View cursor changes to a pointing hand.

▼ To Display a Detailed Picture of the Component

♦ **Click the component with the “detail” pointer.**

For example, if the pointer appears on a CPU panel in the chassis, and you click the CPU, Solstice SyMON displays another Physical View window with a detailed picture of the CPU or memory board (see Figure 3-1).

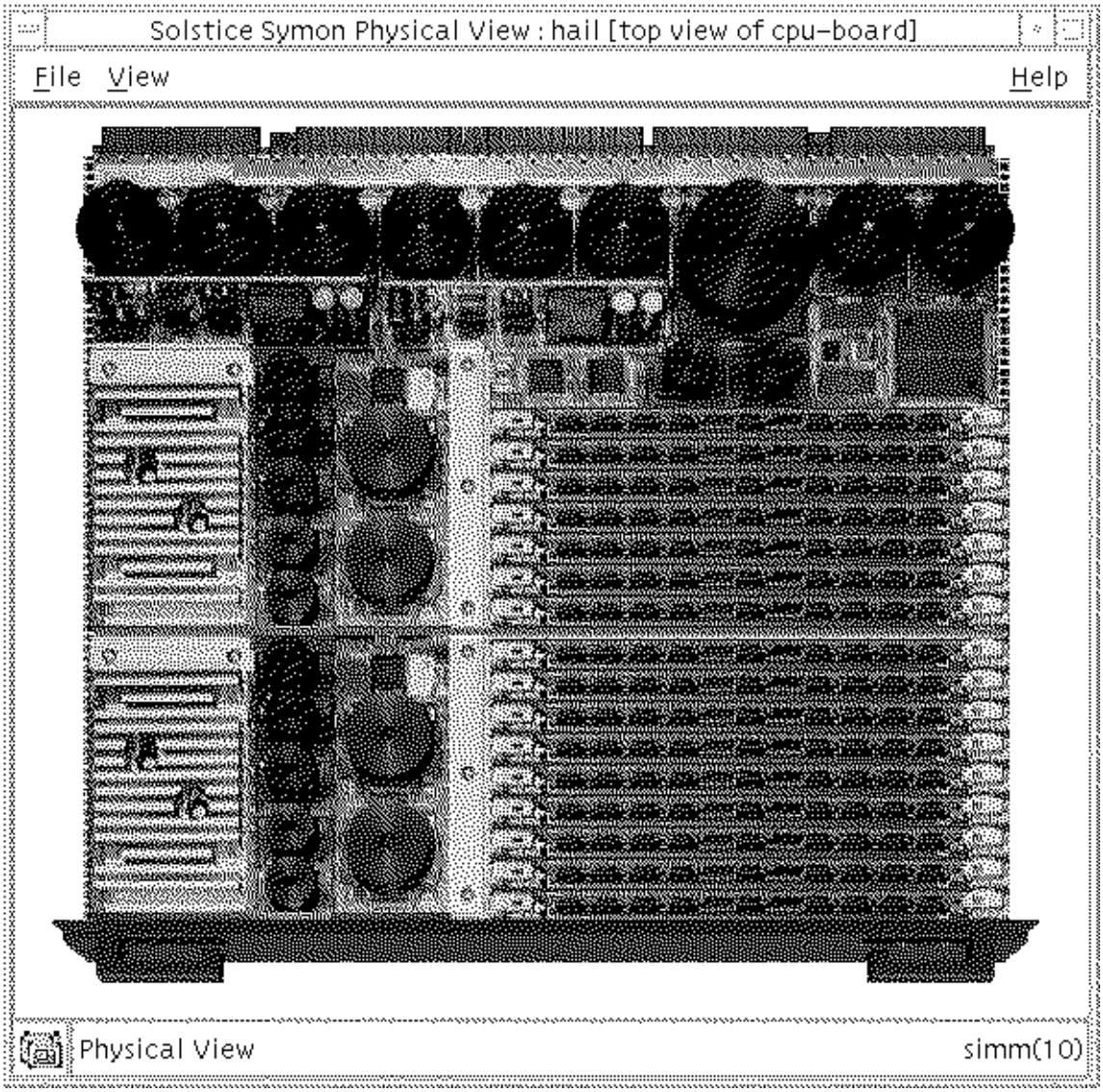


Figure 3-1 The Expanded Physical View

Show Information Option

The View Menu on the main Physical View and any sub-views of individual components include a Show Information option. You can use this option to view more specific information about each component of the system.

When you choose the Show Information option, Solstice SyMON displays a pop-up window. As you move the pointer to a component displayed on the Physical View, information about the component is displayed in the window (see Figure 3-2).

▼ To Use the Show Information Option

- **Place the pointer on the View menu and select the Show Information option.**
  If the Show Information option is active and you place the pointer over a CPU displayed on the Physical View, Solstice SyMON displays critical information about the CPU, including: CPU type, status, cache-size, board number, port ID, processor ID, and model number.

You can display a particular Show Information display on your screen or save it to a file.

▼ To Save the Properties of the System Displayed in the Show Information Window

- **Select the Save As option under the File menu of the Show Information window.**

▼ To Close the Show Information Window

- **Select the Close option on the Show Information File menu.**

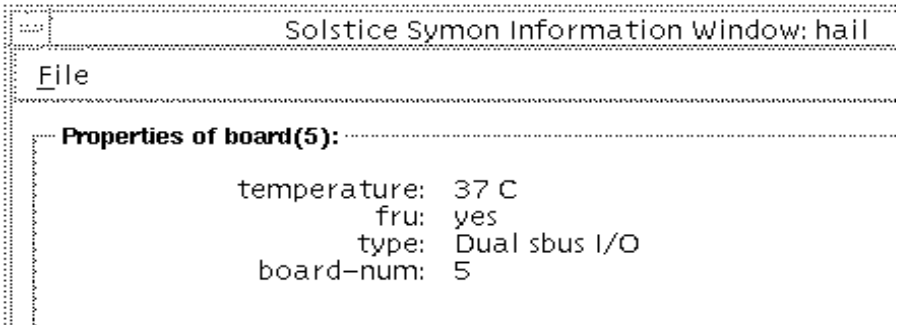


Figure 3-2 Using the Show Information Option

Close Options

▼ To Close a Physical View Using the File Pull-down Menu

♦ **Select any of the following options:**

- Close Self: Use this option to close this window only.
- Close Self and Descendants: Use this option to close this window and any more detailed component displays opened from this window.
- Close All: Use this option to close all Solstice SyMON Physical View windows.

▼ To Quickly Locate Troublesome Components

♦ **Glance at the Physical View and trace events from the top level of the hierarchy down to the component that generated the event.**

Using the Logical View

The Logical View is a companion view for the Physical View. While the Physical View displays the components of the system as physical entities, the Logical View displays the components in a schematic hierarchy (see Figure 3-3).

To show a more or less detailed listing of components, expand or contract branches of the hierarchy (for example CPU, I/O).

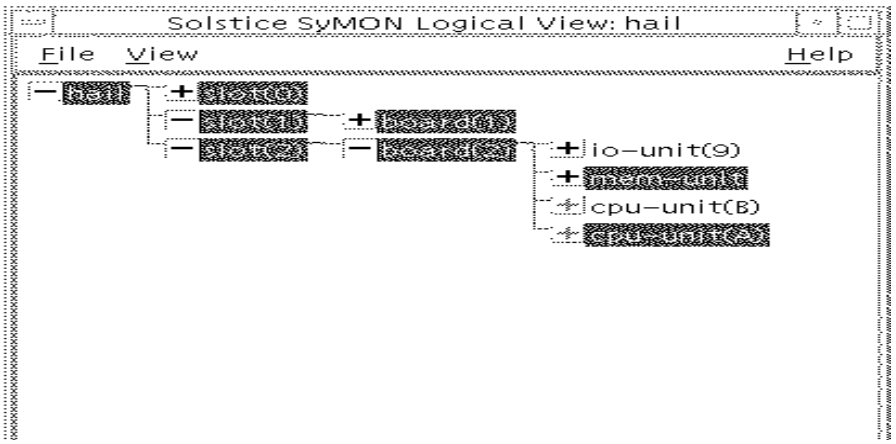


Figure 3-3 The Logical View

For information on how the Event Handler locates troublesome components, see the section, “To Quickly Locate Troublesome Components.”

Expanding and Collapsing the Hierarchical Diagram

Solstice SyMON displays a two-state button to the left of each component in the hierarchy (see Figure 3-3). The two possible states are collapsed (+) and expanded (-).

- Collapsed(+)—Means that you can expand the view to display the next level of the hierarchy (only if it is not grayed out).
- Expanded (-)—Means that the display is already expanded.

Clicking the Collapsed(+) or Expanded (-) button expands or contracts the entire hierarchy below that point one level at a time.

▼ To Expand the Hierarchy

♦ **Click the (+) button in the collapsed state.**
The hierachy below that point is expanded. As you expand the hierarchy, you may need to resize the console to view the entire hierarchy.

▼ To Hide the Contents of the Hierarchy

♦ **Click the button in the expanded (-) state.**
Solstice SyMON hides the contents of the lower level of the hierarchy.

Show Information Option

The View Menu on the Logical View includes a Show Information option that is identical to the Physical View Show Information View Menu.

Use this option to view more specific information about each component of the system. Refer to the Show Information description in the section “Show Information Option” in this chapter.

Using the On-Line Diagnostics Console

The Diagnostics icon launches the Sun Validation Test Suite (SunVTS). SunVTS is a suite of diagnostics that tests and validates the configuration and functionality of most hardware devices of a server.

For more detailed information about SunVTS, refer to the *SunVTS 2.0 User’s Guide*.

Monitoring System Software Performance

Use Solstice SyMON to monitor software performance and to identify system bottlenecks. Chapter 1, “Solstice SyMON Overview” covers how to isolate and identify system bottlenecks in the “Monitoring System Software Performance” section.

3

Using the Kernel Data Catalog to Build System Meters

The Kernel Data Catalog console displays the system performance parameters. These parameters are displayed as “instruments” within the Kernel Data Catalog. These parameters are arranged in a hierarchical diagram by performance category (CPU, memory, disk, network, and process), much like the Logical View shows a hierarchical diagram of the hardware of the system. You can choose whether the instruments associated with each hierarchy are displayed or hidden (see Figure 3-4).

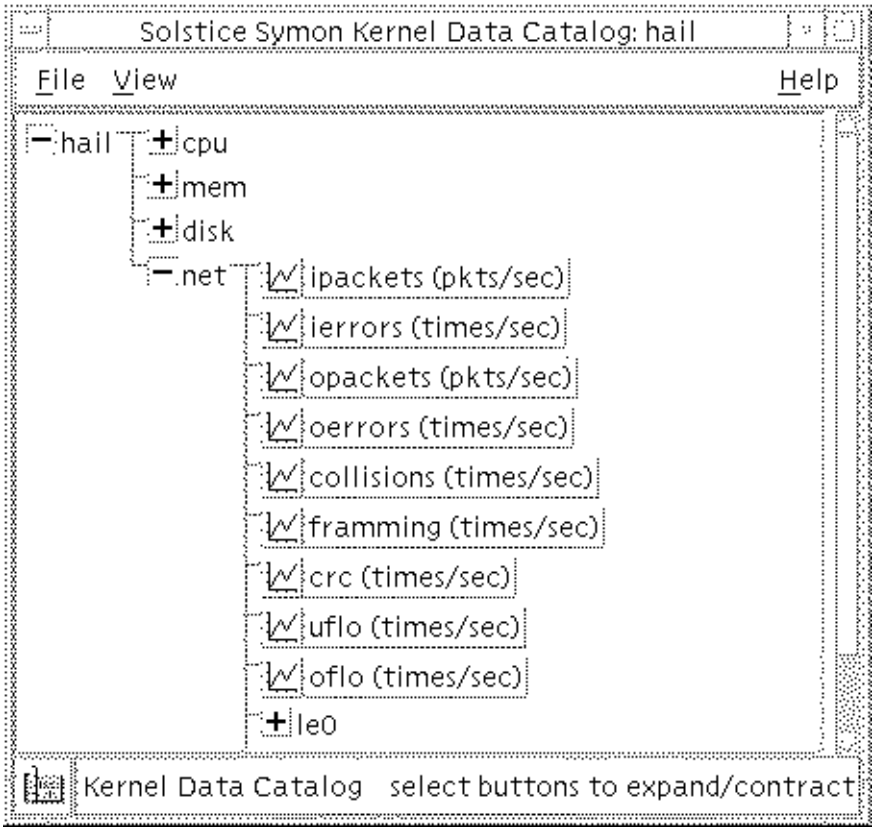


Figure 3-4 The Kernel Data Catalog

When the status of a parameter results in an event, the Event Handler highlights the parameter on the Kernel Data Catalog. If a parameter is highlighted, its ancestors are also highlighted.

For information on how the Event Handler locates troublesome components, see the section “To Quickly Locate Troublesome Components.” To expand or collapse the hierarchical diagram, see the section, “Expanding and Collapsing the Hierarchical Diagram” and the procedures “To Expand the Hierarchy” and “To Hide the Contents of the Hierarchy” earlier in this chapter.

Hiding and Showing Instruments

The View menu of the Kernel Data Catalog console includes an option that toggles between Hide Instruments and Show Instruments. Use this feature to hide or show instruments throughout the hierarchy.

Building System Meter Windows

The System Meter windows help to build a graphical display of system performance parameters.

Use the parameter instruments displayed in the Kernel Data Catalog to:

- Create a graph for one parameter in a System Meter
- Combine multiple parameters on the same graph in a System Meter
- Combine multiple graphs in the same System Meter

▼ To Create a System Meter

1. **Ensure that the instrument for the parameter to be monitored is visible in the Kernel Data Catalog window.**
2. **Place the pointer over the desired parameter and click the left mouse button.**
   Solstice SyMON creates a System Meter window with a graph of the selected parameter.

▼ To Add Another Parameter to an Existing Graph in a System Meter

♦ **Place the pointer over the parameter you want to add and drag and drop it (using the middle or left mouse button, depending on your mouse setup).**
Solstice SyMON adds the new parameter to the existing graph and legend (see Figure 3-5).

▼ To Add a New Graph to an Existing System Meter

♦ **Place the pointer over the parameter you want to add and drag and drop it in the footer of the System Meter.**
Solstice SyMON adds a second pane to the System Meter and draws the graph for the new parameter in the new pane.

Note – One graph has a slightly darker background than the other. The darker one is selected. Only one graph at a time per System Meter can be selected by clicking the pointer on the graph to select.

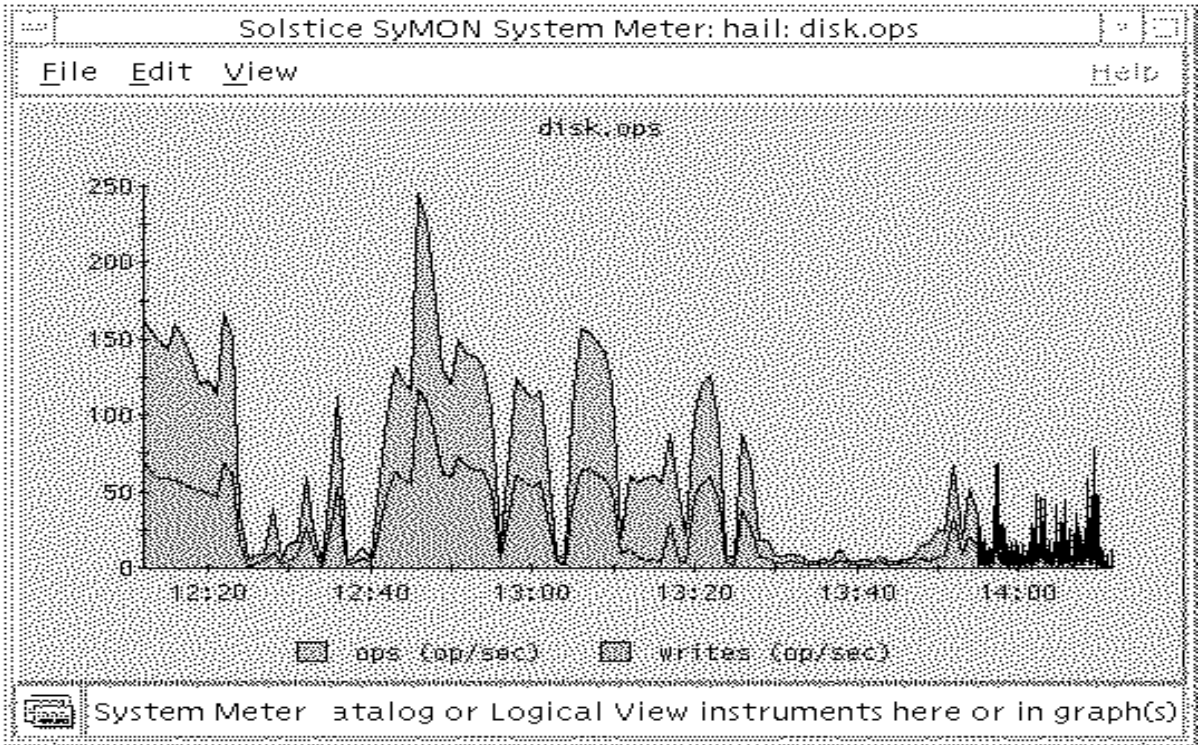


Figure 3-5 Adding a Parameter to an Existing System Meter

Customizing a System Meter

You can use the following options under the View menu to customize a System Meter (see Figure 3-6).

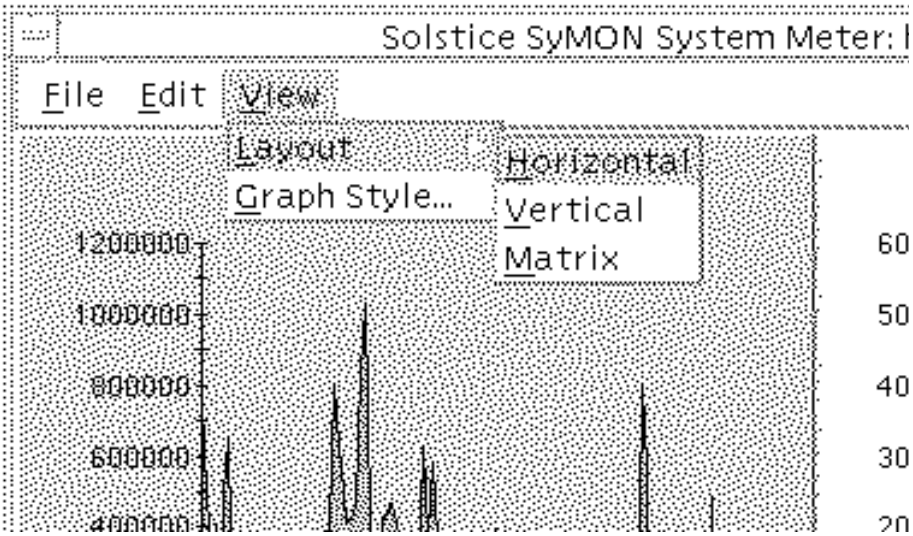


Figure 3-6 Graph View Menu Options

Graph Layout

The Graph Layout option controls the appearance of a System Meter that contains multiple graphs:

- Horizontal: Displays graphs horizontally, from left to right
- Vertical: Displays multiple graphs vertically, from top to bottom
- Matrix: Displays multiple graphs in a grid format

Graph Style

When you select Graph Style from the View pull-down menu, the Graph Style Dialog window appears (see Figure 3-7).

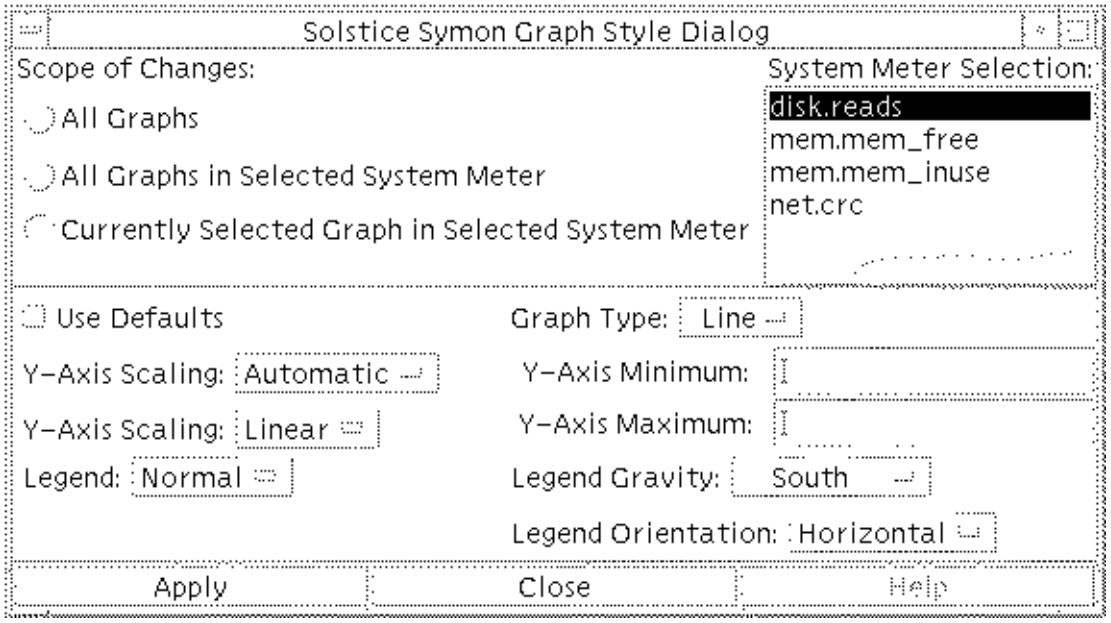


Figure 3-7 Graph Style Window

Modifying the System Meter Graphs

If you selected the “All Graphs” field in the style dialog window, the “All Graphs in Selected System Meter” field becomes inactive.

▼ To Modify the Selected Graph

1. **Select the “Currently Selected Graph in Selected System Meter” field.**
2. **Use the bottom half of the Graph Style form to change the appearance of the graphs you selected.**

Table 3-1 shows the graph attributes you can modify.

Table 3-1 Description of Fields on the Graph Style Form

Field	Option	Description
Use Defaults	Active / Inactive	Selects default settings for all fields at the bottom of the form
Y-Axis Scaling	Automatic	Makes Y-axis scaling changes to reflect changing values (default)
	Fixed	Makes Y-axis scaling constant, regardless of changes in values
Y-Axis Scaling	Linear	Makes Y-axis scaling linear (default)
	Log	Makes Y-axis scaling logarithmic, useful when a graph includes two parameters with vastly different values
Legend	Normal	Shows the legend (default)
	None	Omits the legend
	Long	Extended label for each variable. For example, `cpu.cpu0.busy` (%).
Graph Type	Area	Draws an area graph
	Line	Draws a line graph (default)

Table 3-1 Description of Fields on the Graph Style Form *(Continued)*

Field	Option	Description
Y-Axis Minimum		Sets the minimum value for the Y-axis; used only if Y-Axis is "Fixed"
Legend Gravity	North NorthEast East SouthEast South NorthWest West	Top of the graph Right of the graph Bottom of the graph (default) Left of the graph
Legend Orientation	Vertical Horizontal	Displays the legend in a vertical format Displays the legend in a horizontal format (default)

Changing Selected Graphs

▼ To Change a Graph

1. **Choose the Graph Styles option from the View menu of any System Meter.**
2. **Use the top section of the Graph Styles form to select the graphs that you want to change.**
3. **Make the desired changes to the controls on the bottom half of the Graph Styles form.**
4. **Click the Apply button.**

▼ To Delete a Graph

1. **Select a graph.**
2. **Select Delete from the SysMeter Edit menu.**
   The selected graph is deleted.

▼ To Undo a Change in a Graph

♦ **Select Undo from the SysMeter Edit menu.**

▼ To Delete a Variable from a Graph

1. **Place the pointer over the legend for the variable and click the LEFT mouse button.**
   A dialog box asks, “Do you want to delete the following data set” with the following information contained within the dialog box:
   - System Meter: Variable name
   - Graph: Defines parameters being displayed
   - Data Set: Lists the operation being monitored
2. **Delete the variable by pressing the YES button.**

Displaying the System Meter Graphs

You can move, rotate, scale, or zoom the contents of your graph to better display the information with the following procedures:

▼ To Move a Graph

1. **Position the pointer within the graph.**
2. **Press and hold the Shift key on the keyboard and press and hold the middle mouse button.**
3. **Move the graph using the mouse.**

▼ To Scale a Graph

1. **Position the pointer within the selected graph.**
2. **Press and hold both the Control key then hold down the middle mouse button.**
3. **Move the mouse to scale the graph and release the mouse button to stop the graph motion.**

▼ To Zoom a Graph

1. **Press and hold down the Control key and the Shift key, and the left mouse button at one corner of the graph.**
2. **Drag the mouse to form a rectangle around the area you want to magnify, while holding down the middle mouse button.**
3. **Release the mouse button to define the rectangle and to complete the zooming action.**

Note – If you release the Control or Shift keys too early, the graph starts a drag-and-drop operation instead of zooming.

▼ To Restore a Graph to the Default Setting

- **Position the cursor within the desired graph and type** `r`.

Saving and Redisplaying System Meter Configurations

You can save and recall System Meter layouts and configurations you have defined. For example, you may have a specific set of CPU monitoring graphs that you would like to view every time your run Solstice SyMON.

▼ To Save the System Meter Configuration into a New File

1. **Choose the Save As option from the View menu.**
   The software displays the File Selection Dialog box.
2. **Select the directory and file name from the Directories and Files panes or enter a new file name in the Selection pane.**

▼ To Save an Altered System Meter Configuration

- **Choose the Save option from the File menu.**
  Solstice SyMON writes the SysMeter contents to the existing file and displays the name of the saved file in the footer of the System Meter window.

▼ To recall a saved System Meter

1. **Choose the Open option from the File menu.**
   The software displays the File Selection Dialog box.
2. **Select the file name from the Directories and Files panes or enter the file name in the Selection panel.**

Note – You can save or restore sets of System Meters by using the File pull-down menu in the Kernel Data Catalog.

Using the Log Viewer

Use this console to search the system log file for entries that occur at a specific time or for entries that contain a specific keyword, or word pattern. Opening the Log Viewer console displays the Log File Viewer window as shown in Figure 3-8.

Note – A new Log File Viewer window is opened each time you select the Log Viewer console button on the Launcher. This allows different criteria to be monitored at the same time.

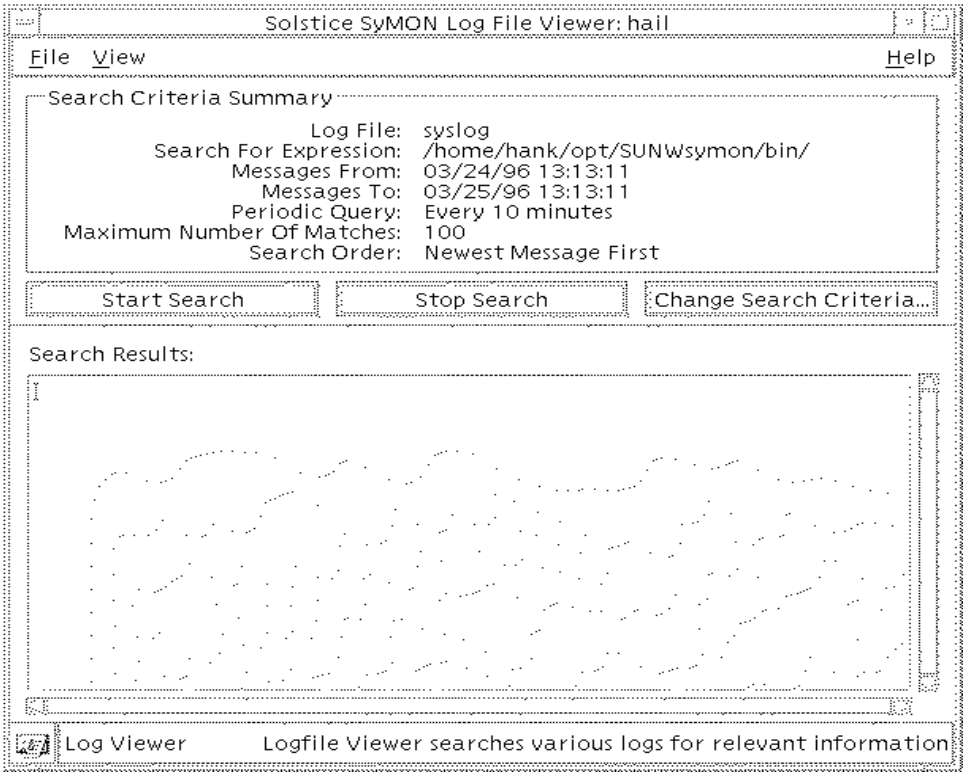


Figure 3-8 The Log File Viewer Window

The View pull-down menu displays the following:

- Log File Status: Describes the Log Scanner status
- Show Search Criteria Summary: Toggle button that displays or hides the Search Criteria Summary
- Display Oldest / Newest Message First: Defines the sort criteria displayed in the Search Result window; this option can be toggled to display either the Oldest message first or the Newest message first

Searching for Log Entries

Initially, the Search Results window will be empty until you specify the Search Criteria and initiate the search. You can specify your own Search Criteria, retrieve and use a stored Search Criteria, or retrieve and modify a stored Search Criteria as defined further in this section.

Log entries retrieved by a search includes the items listed in Table 3-2.

Table 3-2 Search Criteria

Search for	Description
Log File	Name of the log file viewed
Search for Expression	User-defined keyword, string, or regular expression
Messages From	Start date and start time of search
Messages To	End date and end time of search
Periodic Query	interval (in minutes) between repeated searches
Maximum Number of Matches	Only this number of messages is to be reported
Search Order	Oldest (or newest) message searched first

If Solstice SyMON locates any entries in the selected log file that meet the search criteria, the entries are displayed in the Search Results pane. To change the order of the information you gathered in your search, proceed as follows:

▼ To Sort the Results of the Search

- **Select either the Display Oldest Message First or the Newest Message First toggle button in the View menu to display messages in the desired search order.**

▼ To Change the Search Criteria

- **Select Change Search Criteria from the Log File Viewer window.**

▼ To Initiate a Search

- **Select Start Search from the Log File Viewer window.**

▼ To Stop a Search

♦ **Select Stop Search from the Log File Viewer window.**

Note – Due to the client-server structure of Solstice SyMON, the search may appear to continue for several seconds. This will be more evident when the server is fully loaded.

▼ To Save the Search Results

♦ **Select Save Search Results from the File pull-down menu.**

▼ To Save the Search Criteria

♦ **Select the Save Search Criteria option in the File pull-down menu.**

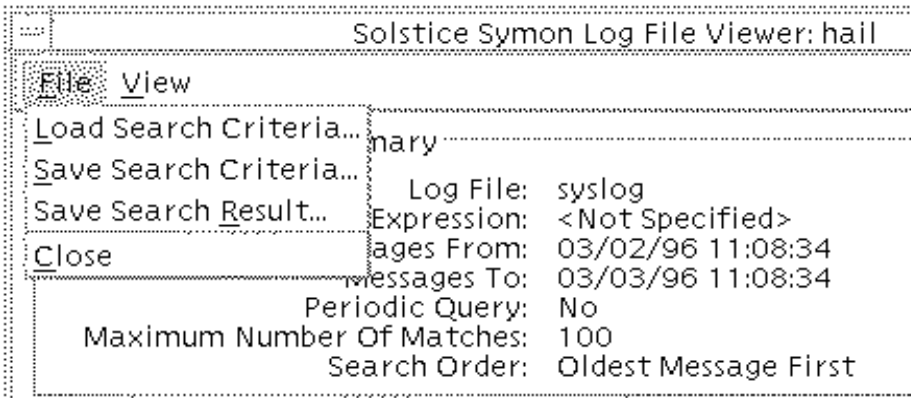


Figure 3-9 Log File Viewer Pull-down Menu

Searching for Log File Entries

You can define your search criteria for specific log file entries by selecting the Change Search Criteria button in the Log File Viewer window.

▼ To Specify the Start Date, Start Time, End Date, and End Time

1. **Move the pointer to the "From" field text field.**
   Enter the time in 24-hour format (see Figure 3-10). The "From" time defaults to when the log was created.

2. **Move the pointer to the "To" field text field.**
   Enter the time and date field. The "To" time defaults to the current time.

▼ To Search for Log File Entries that Contain a Specific Expression or Keyword

1. **Move to the Search for Expression field.**

2. **Enter the expression or keyword.**

   You can also perform more complex searches by specifying a `grep`-like regular expression instead of a keyword. For regular expression syntax, refer to the `grep` man page.

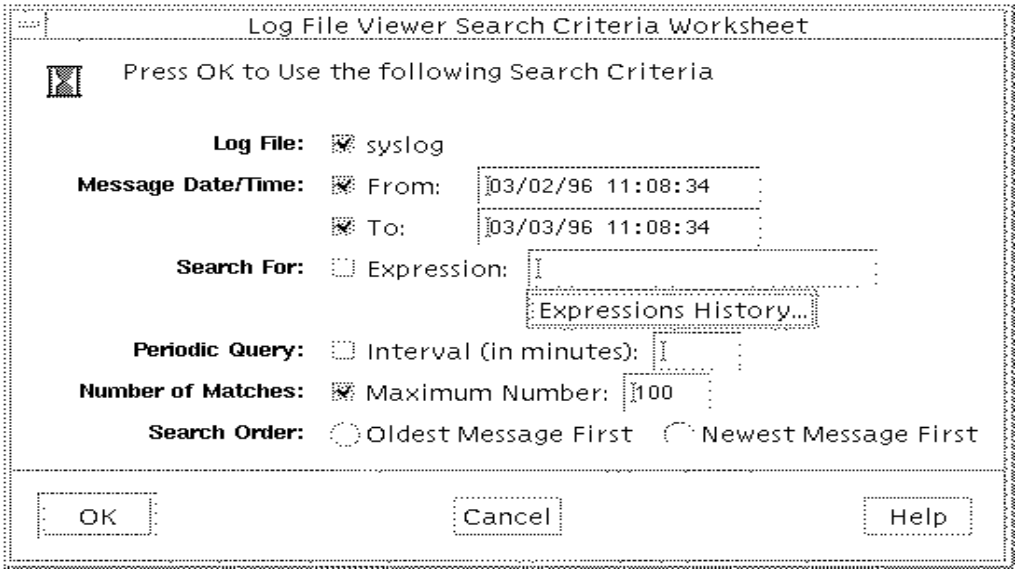


Figure 3-10 Log Viewer Change Criteria Worksheet Window

The Worksheet entries can be modified, or new entries can be entered into the desired search parameters, to include:

- Searches for a specific keyword or expression
- Events that occur during a specific time and contain a specific keyword
- Periodic monitoring of specific events
- Maximum number of matches of these search criteria to be reported
- Listing the messages in ascending or descending order of occurrence

Using the Process Viewer

The Process Viewer displays information on processes running on the server (see Figure 3-11). Each entry displayed on this console represents a process with some of the following data:

- When the process was started
- Who started the process
- How much CPU time was used by the process
- Current process status

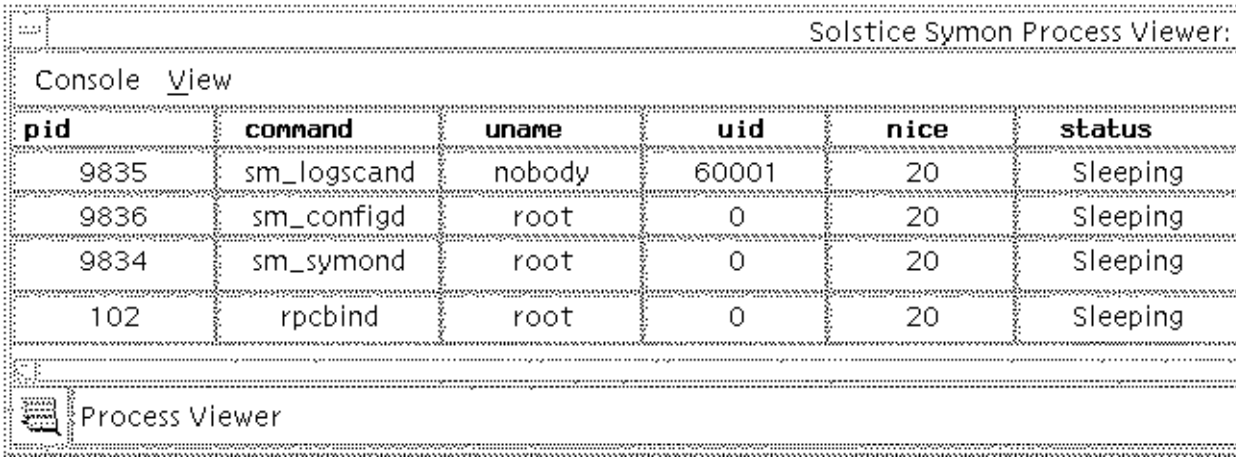


Figure 3-11 The Process Viewer

Monitoring Resource-Intensive Processes

Resource-intensive processes such as compiles, simulations, and complex database manipulations can slow server performance. The Process Viewer console gives you detailed information about resource-consuming processes that run on your server. This information can help you identify resource hogs that can contribute to service problems.

Processes that are inactive or are consuming little CPU time are not reported. This feature eliminates the need to search through dozens of relatively idle processes to view data on resource-intensive processes. In fact, a process may appear and disappear from this console as its resource consumption changes.

Table 3-3 describes the information displayed in each Process Viewer entry:

Table 3-3 Description of Entries on Process Viewer

Field	Description
`pid`	Process ID
`command`	Name of program
`uname`	User name
`uid`	User ID
`nice`	Determines when the process can voluntarily lower its priority. As the `nice` value increases, the process is more willing to give up CPU (value defaults to 20). For more information, see the `nice` man page.
`status`	Lists the current status of the process: - Sleeping: Not generating any activity - Running: Generating activity - Runnable: Waiting for I/O - Trace: A debugger is running
`user (%)`	Amount of CPU time that the process is using
`sys (%)`	Amount of CPU time the kernel is using to perform tasks

Table 3-3 Description of Entries on Process Viewer *(Continued)*

Field	Description
`Total (%)`	Percentage of total CPU time used over the entire life of the process
`Total CPU (%)`	The combined seconds and percentage of CPU time used by the system, user, and short-lived processes
`Start up time`	Time the process started; if the process has been running for more than 24 hours, it displays the date the process started

Customizing the Process Viewer Display

You can customize the Process Viewer window to suit your specific display requirements.

▼ To Organize the Process Viewer Window

♦ **Use the options under the View menu to organize the Process Viewer.**

Sorting

The View pull-down menu options are:

- Sort: The menu displays a list of the fields (columns) for each process view. You can choose one of these fields as the basis for sorting the process views. You can also click on the column's title bar to change the sort order.
- Ascending / Descending Sort Order: The current direction (ascending or descending) of the sort is displayed; if you click the option, it will toggle to the opposite direction.

Adjusting Column Widths

You can also resize the columns.

▼ To Resize the Column

1. **Place the pointer in the column that you want to resize.**

2. **Move the pointer to the left or right margin of the column.**
   The pointer changes from a cross hair to an arrow with a line to right ⇥
3. **Press and hold the left mouse button.**
4. **Drag the margin to the desired spot, and release the mouse button.**

Advanced Features

You can set the order in which columns are displayed, their default width, and the default sort column through the `common.tcl` file, which is read when Solstice SyMON starts. See the "Customizing the Appearance of the Event View and Process View" section at the end of this chapter.

Monitoring Alarms and Events

The Event Handler monitors data from the server and generates events when anomalous conditions occur on your server. The condition, and their corresponding alarms, are described by a set of event rules. See Chapter 4, "Understanding and Using Event Rules".

Using the Event Viewer

If the Event Handler senses an event, it adds a description of the event to the Event Log (see Figure 3-12). The Event Log is displayed via the Event Viewer.

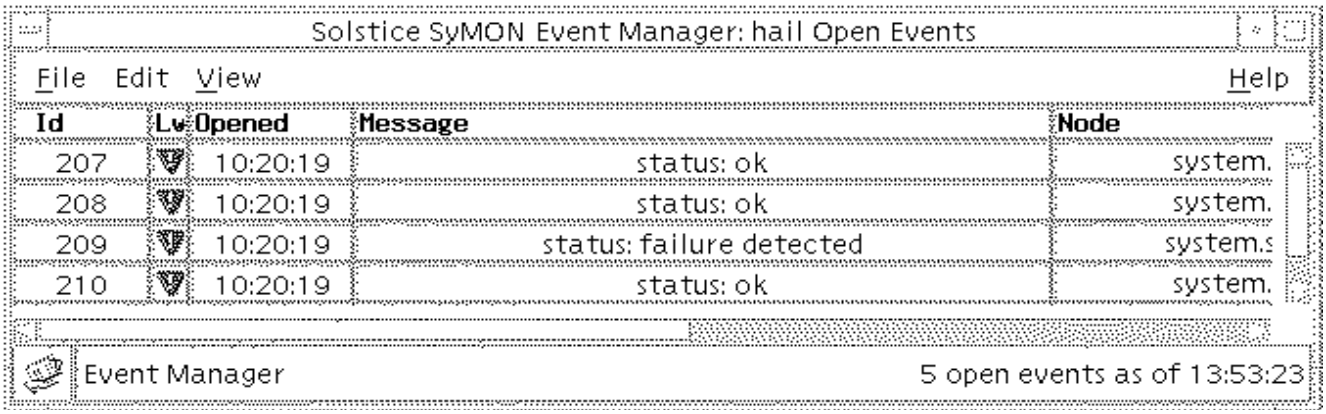


Figure 3-12 Event Viewer

Table 3-3 displays the information found in each event log entry. Some information is optional.

Table 3-4 Description of Entries in Event Viewer

Field	Description
ID	Sequential number assigned to the event
Lvl	Level of events: yellow (caution), red (danger)
Opened	Time the event occurred
Message	Event message (failure detected, for example)
Node	Where the event was detected
Pri	Priority of the event rule
Sev	Severity level of the event
Acked	Time a user acknowledged the event
Ack By	User ID of person acknowledging the event
Rule ID	Number of the rule that caused the event

Two displays are available from the Event Viewer:

- Open Events: Displays only currently open events. This is the view you see when you select the Event Viewer icon from the launcher.
- All Events: Displays all events (even if closed), until they are explicitly deleted.

When an event is open, the Event Handler highlights the event in the Logical View and Physical View or the Kernel Data Catalog until the cause of the event is corrected.

When the condition that caused an event ends, the description of the event is removed from the Open Event view and any associated highlights in the consoles are also removed. The Event Handler notes the closing of the event in the event log.

Acknowledging and Deleting Events

When you acknowledge an event, it has no effect on the event condition or alarms. However, it informs all Solstice SyMON users that the event has been noticed.

▼ To Acknowledge an Event

1. **Select the event to be acknowledged.**
2. **Select Acknowledge from the Edit pull-down menu.**
   The Event Handler enters the time and date in the Acked column and your User ID in the Ack By column.

Note – If you acknowledge or delete an event, the change appears in all Solstice SyMON GUIs.

After the event is closed, it remains in the All Events view until a Solstice SyMON user deletes it.

▼ To delete a closed event

1. **Select the All Events from the View pull-down menu.**
2. **Select the event to be deleted.**
3. **Select Delete from the Edit pull-down menu.**

Note – You cannot delete an Open event from the event list.

Customizing the Event Viewer

Like the Process Viewer Display, you can customize the Event Viewer window to suit your specific display requirements.

Sorting Event Data

To use the options under the View menu to sort Event Data, see the section “Sorting” earlier in this chapter.

Adjusting Column Widths

To resize the columns, see the procedure, “To Resize the Column” earlier in this chapter.

Customizing the Event Viewer and Process Viewer

The appearance of the Event Viewer and Process Viewer tabular displays of the Event View and Process Viewer are controlled by the file `~/.symon/lib/tcl/common.tcl`. This file is read and interpreted when the Solstice SyMON GUI is started. You can customize the appearance of these consoles by editing this file.

In this file, three major Tcl variables are set: `symon_process_columns`, `event_display_open_columns`, and `event_display_all_columns`. These variables control the data that appears in the display; the order in which they appear; and the default column width for each column.

Within each of these variables is a line for each data item. The order in which these data item lines appear in the Tcl variable is the order in which their respective columns appear in the displays.

A typical line looks like this:

```
{"tpcnt"   "   Total (%)   "     "+"}
```

The first string is the Solstice SyMON internal name for the data item. The second string is the title for the column in which it is displayed. The total length of this string (including spaces) controls the default width of the column. The third string (which can only be “+” or “-”) controls the default sort order of the column (+ for ascending, - for descending). Some lines may have a fourth string with a format specifier such as `%s` or `%d`; these are unused and may be omitted.

Associated with each of the appearance-controlling Tcl variables are two other variables: `event_display_all_columns_alternate` and `event_display_all_columns_sort_node`. The `_sort_node` variable takes a single argument: the name of the data item used to sort the display when it initially appears.

The `_alternate` variable is used to hold control lines for data items that do not appear in the display.

3

Understanding and Using Event Rules

This chapter provides an overview of how to understand and write Solstice SyMON event rules. This chapter describes:

- Event rules terminology
- Tcl rules
- Hierarchies
- Simple rules
- Complex rules
- How to write event rules
- How to verify new event rules
- Reserved words
- Special characters
- Debugging tips

Event Rules Terminology

Events are hardware or operating system conditions that may require the attention of a system administrator. Examples of events include: the loss of a CPU or disk, processor overload, or excessive swapping.

The Event Handler subsystem of Solstice SyMON alerts you to events. However, you must first write a rule that defines the event. *Rules* include a *condition* and other attributes that define the state of the rule and actions to take.

A *condition* is an expression that defines when a rule is active. An example of a condition might be a failed board.

An *action* tells Solstice SyMON what to do when a condition is true, if the condition changes, or if the system shuts down. Actions notify users of a situation that may require attention.

In addition to conditions and actions, a rule may also include other attributes, such as the level, priority, and severity of a rule.

The Event Handler collects data from monitoring agents and evaluates the data against its rules. When a condition is true, the Event Handler logs an event and carries out the appropriate action in the rule. When the condition that generated the event no longer exists, the Event Handler may run a special action such as closing the event. The Event Handler is always running. If it stops and restarts, all previously open events are closed.

Tcl Rules

The event rules are written in the Tcl scripting language. For a definition of Tcl syntax and complete instructions on how to write Tcl scripts, refer to:

- *Tcl and the Tk Toolkit*, by John K. Ousterhout, Addison-Wesley Publishing Co.: 1994.
- *Practical Programming in Tcl and Tk*, by Brent B. Welch, Prentice Hall: 1995.
- Tcl Reference card—available from Specialized Systems Consultants, Inc., P.O. Box 55549, Seattle WA., 98155-0549.

Caution – Do not modify event rules if you do not know Tcl.

For additional information, contact software support at Sun Microsystems, Inc. at 1-800-USA-4SUN or 1-800-872-4786.

Solstice SyMON includes a set of pre-defined rules in the Tcl variable `RULES` located in the `/etc/opt/SUNWsymon` directory. The following rule files are included:

- `rules.tcl` — organizer
- `cprules.tcl`— capacity planning rules
- `hwrules.tcl`— hardware monitoring rules

- `syrules.tcl`— Solstice SyMON monitoring rules
- `egrules.tcl`— Event Handler rules
- `pfrules.tcl`— predictive failure rules
- `swrules.tcl`— system monitoring rules

Writing rules is simplified by using a set of Tcl commands that tell Solstice SyMON what to do in a given situation. For example, the Tcl `alarm` command tells Solstice SyMON what to do to make an event active. The Event Handler reads the rules file.

You may expand Tcl to include your own procedures by editing the `event_gen.tcl` file.

Special Characters

Tcl includes these special characters:

- "(quotes)
- { }(curly bracket)
- [ ](square bracket)
- # (pound sign)
- * (asterisk)

Writing rules is simplified by using a set of Tcl commands and procedures. For more information on these characters, refer to the Tcl reference material in the “Tcl Rules” section.

4

Reserved Words

Table 4-1describes the Tcl reserved variable names, which contain certain values that have meaning for the Event Handler. You cannot redefine these variable names.

Table 4-1 Tcl Reserved Variable Names

Variable Name	Meaning
`symond_status`	Tells whether the Solstice SyMON daemon is running
`server_status`	Tells whether the monitored system is up
`LogScanner_status`	Tells if the Log Scanner agent is running
`ConfigReader_status`	Tells if the Config Reader agent is running
`KernelReader_status`	Tells if the Kernel Reader agent is running
`node`	Contains the hierarchy that is being evaluated in the rule
`myroot`	Root node for the Event Handler hierarchy; it should not be changed by the user

Event Rule Attributes

Each attribute, with the exception of the *condition* attribute has a label followed by a value. The condition attribute does not have a label and is a mandatory attribute for each rule.

Table 4-2 contains a list of attribute descriptions.

Table 4-2 Attribute Descriptions

Name	Value	Description
`RULE`	Integer	Rule number
`ON_OPEN`	Tcl script	Tcl string to be interpreted when the condition of the rule becomes true
`ON_CLOSE`	Tcl script	Tcl string to be interpreted when the condition of a rule is no longer true
`ON_CONTINUE`	Tcl script	Tcl string to be interpreted when the condition of a rule continues to be true

Table 4-2 Attribute Descriptions *(Continued)*

Name	Value	Description
ON_SHUTDOWN	Tcl script	Tcl string to be interpreted when the Event Handler shuts down
PARAMETERS	String	User-defined parameters
MULTI	Tcl script	Tcl string to provide a list of data node variables for multiple events of a single rule
SEVERITY	Integer	Severity of the event (user-defined and interpreted)
PRIORITY	Integer	Priority for the event (user-defined and interpreted)
RATE	Integer	For the user-defined rule sampling rate, in seconds
COMMENTS or C	String	Comments field
LOG_RULES	Log script	Tcl string that defines the log file scanner activity that supports rules

Rule Functions

A rule function is a Tcl command used in Tcl scripts associated with the RULES variable. Table 4-3 lists common commands.

Table 4-3 Common Rule Commands

Command	Description
alarm	Makes an event active, highlights the hierarchy node (RED, YELLOW, or BLUE) and creates entries in both the event log and the Event Viewer (with predefined message) of the event. The syntax is alarm *level node "message" "Tcl command"*.
end_alarm	Closes the log entry in the Event Log; the syntax is end_alarm.
findlist	Builds the list of matching nodes. The syntax is findlist *hierarchypath striplist.*

Table 4-3 Common Rule Commands *(Continued)*

Command	Description
`findvalue`	Takes the name of any data hierarchy variable and returns the value of the variable; the syntax is `findvalue` *node* (where *node* is the hierarchy endpoint).
`getfield`	Returns the internal data value from a rule or a rule-node combination (list from a MULTI string). The syntax is `getfield` *optional_rule_# field_type*. For example, `{ [getfield COUNT] == 1}`. The following field specifiers can be used with `getfield`: - `ACTIVE`: True or false if the rule is currently active - `COUNT`: Returns the number of consecutive iterations of the rule being active - `PRIORITY`: Returns the priority value of the rule - `SEVERITY`: Returns the severity value of the rule - `RULE`: Returns the rule number - `START_TIME`: Returns the time the rule became active
`putfield`	Takes a field name and data and assigns that data to the field of the current rule. The syntax is `putfield` *optionalrule# fieldtype "value"*.
`get_parameter`	Manages a parameter list; used for `MULTI` rules that need to maintain historical records. The syntax is `get_parameter` *string*.
`put_parameter`	Manages a parameter list; used for `MULTI` rules that need to maintain historical records. The syntax is `put_parameter` *tag value*.
`gettime`	Gets the sample time on the monitored machine in a long integer. The syntax is `gettime`.
`dynlink`	Takes a shared object and procedure name and dynamically links the shared object and calls the name procedure. The syntax is `dynlink` *file functionname*.

Table 4-3 Common Rule Commands *(Continued)*

Command	Description
`mailto`	Sends a message to a specified name by email. The syntax is `mailto` *address "msgstring."*
`syslog`	Writes the specified string to the `syslog`. The syntax is `syslog` *"message."*
`snmp`	Initiates SNM traps . `snmp` takes a string as an argument and generates snmp traps on every machine in the `snmp_hosts` variable. `snmp_hosts` is defined in `event_gen.tcl`. The syntax is `snmp` "*message.*"

Hierarchies

Hierarchies organize data in Solstice SyMON to handle the grouping of related pieces of information. A hierarchy has a:

- Top level node representing all data from an agent
- Subset of the top level node for classes of data
- Subnode for closely related data and properties containing the actual data

Each node and subnode organize the data beneath it. For example, in:

```
KernelReader.cpu.cpu1.busy
```

- The top level `KernelReader` indicates that this is KernelReader data.
- `KernelReader.cpu` indicates that this is CPU data.
- `KernelReader.cpu.cpu1` indicates that this data is related to CPU1.
- `KernelReader.cpu.cpu1.busy` is the percentage of time that the CPU1 was busy.

For more information on what data is available, refer to the Solstice SyMON man pages. For detailed information on the Kernel Reader, see Appendix A, “Kernel Reader.”

The following sections present examples of simple and complex rules.

Simple Rules

A simple rule checks one or more variables in a simple condition and generates one event if the condition is true. Code Example 4-1 is an example of a simple rule.

```
{
        RULE 2
        { expr { "$server_status" == "dead" } }
        ON_OPEN { alarm RED "" "Server not responding" "" }
        ON_CLOSE { end_alarm }
        SEVERITY 1
        PRIORITY 1
}
```

Code Example 4-1 Simple Rule Example

The first attribute in this rule of Code Example 4-1is the rule number.

The second attribute is the condition. This condition checks to see if the `server_status` variable is equal to “dead.” If `server_status` is equal to “dead,” the rule is active. `expr` is a Tcl function that evaluates an expression and returns a value. Tcl variables such as `$server_status` in Code Example 4-1 are part of the Event Handler variables.

The third and fourth attributes are a set of actions that are carried out as the conditions of the rule change. When the rule becomes active, the rule triggers an alarm with the `RED` condition. The predefined Tcl `alarm` command activates an event and writes an entry into the Event Log. When the condition is no longer true (`ON_CLOSE`), the rule triggers `end_alarm`, which closes the log entry in the Event Log, removes any highlighting in the GUI, and deletes the event from the open event list.

The `SEVERITY` and `PRIORITY` of the rule are the last two attributes in the rule. These are numeric values defined by the user.

The following rule in Code Example 4-2 is slightly more complicated. It defines a swap space event:

```
{
        RULE 18
        {
         set ts [ expr 0.10 * [ findvalue \
KernelReader.mem.swap_total ] ]
       expr { [findvalue KernelReader.mem.swap_free ] < $ts }
        }
    ON_OPEN { alarm RED KernelReader.mem.swap_free "Serious Swap
Problem" "" }
        ON_CLOSE { end_alarm }
        SEVERITY 2
        PRIORITY 1
}
```

Code Example 4-2 Rule 18: Monitoring Swap Space

The first attribute in this rule is the rule number.

The next attribute is the condition. The condition does the following:

- Sets variable `ts` (total swap space) to 10 percent of the total swap space available on the machine. `KernelReader.mem.swap_total` is a performance property in the data hierarchy of Solstice SyMON; `findvalue` is a predefined Tcl command that returns the value of the performance variable
- Finds out how much swap space is free and unused
- Compares the values for unused swap space and total swap space; if the unused swap space is less than 10 percent of total swap space, there is a potential problem

The `ON_OPEN` attribute tells the Event Handler to generate a `RED` alarm by calling the alarm function with the `RED` argument. This highlights a node on the Kernel Data Catalog and adds the event to the Event Log. All the arguments for the `ON_OPEN` alarm function attribute are mandatory. If an argument is not used, it is replaced by a set of double quotes ("").

The ON_CLOSE attribute tells Solstice SyMON what to do when the condition becomes false; end_alarm closes the log entry in the Event Log and the event in the Event Viewer.

Note – Always explicitly close an alarm with the end_alarm function. An alarm does not automatically close when the condition goes away.

The SEVERITY and PRIORITY of the rules are the last two attributes. These are both user-defined and interpreted.

Complex Rules

A complex rule checks the condition against several hierarchy nodes. If any condition is true, it generates an event for that variable.

Complex rules eliminate the need to write many simple rules that check the same condition. For example, if you use simple rules to check the condition of each CPU on a server, you will write many simple redundant rules.

Code Example 4-3 is an example of a complex rule:

```
{
       RULE 1
        MULTI { expr { [ findlist system.*.*.*.status "" ] } }
        {
                set boardstatus [ findconfigvalue $node ]
                expr { "$boardstatus" == "failure detected" }
        }
        ON_OPEN { alarm RED $node "Board failure detected" "" }
        ON_CLOSE { end_alarm }
        SEVERITY 1
        PRIORITY 1
}
```

Code Example 4-3 Complex Rule Example

Rule 1 examines the status of all boards in a system in Code Example 4-3. The condition of the rule is that the board has failed. If the board fails, the Event Handler opens a `RED` event and sends a message to the Event Log (`ON_OPEN`). When the condition is no longer true (`ON_CLOSE`), the Event Handler executes the `end_alarm` procedure.

The `MULTI` attribute is a Tcl script that evaluates an expression and creates a list of nodes. The condition of the rule is run once for each node in the list, and the Tcl node variable is assigned the value of the node being processed or evaluated. Any other Tcl script associated with the rule can access this value with `$node`. The normal approach is to use the `findlist` function to generate a list of nodes and check their values in a rule. A `findlist` must be present in a `MULTI` attribute.

The following statement, taken from Code Example 4-3, generates a list of all data items that start with `system` and end with `status`.

```
MULTI { expr { [ findlist system.*.*.*.status "" ] } }
```

Writing Event Rules

You can create rules against data from any of the three Solstice SyMON agents: Config Reader, Kernel Reader, and Log File Scanner. The first two agents provide data continuously to the Event Handler. You only need the path name to the variable.

For the Log File Scanner, you must pre-define the messages that are sent to the Event Handler for examination.

Here are a few guidelines to keep in mind when writing event rules:

- A rule is a Tcl variable.
- The Event Handler rule attributes and labels are strings to Tcl.
- A Tcl variable is a string or list of strings.
- Rules are free form. They only require attributes (label and data) pairs and can appear in any order.
- Attributes are separated by new lines or spaces.

- Each attribute consists of one or two components. The first component is the name (label) and the second component is the associated value. The exception to this rule is the condition, which does not have a label.
- An attribute's components are separated by spaces, tabs, or new lines.
- Only the last instance of each label is used per rule. All others are ignored.
- Spaces are allowed in an attribute item data if it is enclosed in quotes.
- Each label may have one unique item.
- Each Tcl command string is enclosed in curly brackets.
- A condition is the only mandatory attribute in a rule.
- A condition of a rule can look at time or other values in the environment. For example, "`set current_time [gettime]`" sets the user-defined variable, `current_time`, to the current time on the monitored machine.
- A rule does not have to include an action.
- A rule can set a value that is used in other rules.
- All rules are combined into a single Tcl variable called RULES.

Solstice SyMON can execute `phone_home` scripts.

Verifying New Event Rules

Solstice SyMON includes the special `verify_rules` command, which checks for Tcl syntax. The command takes three optional arguments, which are described in Table 4-4.

Table 4-4 `verify_rules` Arguments

Argument	Description
`-R`	Checks the file that contains the EVENTS variable; the default is `rules.tcl`
`-I`	Checks the file that contains supporting functions; the default is `event_gen.tcl`
`-o`	Gives more verbose output

♦ **To verify rules, enter:**

```
$ verify_rules filename
```

The *filename* entry is optional. The default *filename* is `rules.tcl`.

When you run `verify_rules` and the rules are correct, the program responds, "`GOOD RULES`." If the program detects a syntax error, the program responds, "`BAD RULES`." This does not guarantee that the rule will work as written.

Activating New or Modified Events

To activate new or modified rules, send the signal `SIGHUP` to the Event Handler or restart the Event Handler. Send `SIGHUP` by invoking the following command:

```
% kill -HUP pid
```

where `pid` is the process ID number of the Event Generator, `sm_egd`. For more information, see the `kill` man page.

Debugging Tips

Use the following debugging tips for event rules:

- Run `verify_rules` to make sure the syntax is accurate.
- If there is a problem with the rules, change the `rules.tcl` file so you test only one rule at a time.
- Use this event log file to search for error messages: `/var/opt/SUNWsymon/`*machine_name*`/event_log`.
- Use the `syslog` command to log variable values and to confirm the values.

Kernel Reader A

This appendix provides a summary of the Kernel Reader data hierarchy.

Kernel Reader Data Hierarchy

Table A-1 lists the Kernel Reader data hierarchy. The first column identifies whether the parameter is a node (N) or a property (P). A node in the hierarchy is a starting point for a branch of the hierarchy, similar to the point where a branch meets the trunk of a tree. All properties that belong to a node appear below the node and are marked with a P. For example, the `cpu.busy` and `cpu.user` properties fall under the `cpu node`.

The second column contains the name of the node or property.

The third column contains two values. The first value is a two-prong value set off in parentheses, such as (`R.f`). This value identifies the type of value and the data type.

Solstice SyMON generates five types of values:

- (I) Instant value at the sample time
- (S) Summary over the last sample or report interval
- (A) Average value over last sample or report interval
- (%) Percentage value as float (100.0 == 100%) over the last sample or report interval
- (R) Rate (per second) value over the last sample or report interval

Note that there is no value type or data type for nodes. Data types include `f` (float), `i` (integer), and `s` (string [text]).

The second value in the third column is a brief description of the node or property.

Table A-1 Kernel Reader Data Hierarchy

Node or Property?	Name of Node or Property	Value or Data Type and Brief Description
N	`KernelReader`	Overall root
N	`control`	Control information
P	`control.time`	(`I.i`) time stamp of report
P	`control.pid`	(`I.i`) process ID for the Kernel Reader
N	`cpu`	CPU usage over all CPUs
P	`cpu.busy`	(`%.f`) CPU busy time (usr+sys,100.0 == 100%)
P	`cpu.user`	(`%.f`) user time
P	`cpu.sys`	(`%.f`) sys time
P	`cpu.wait`	(`%.f`) wait time
P	`cpu.idle`	(`%.f`) idle time
P	`cpu.ncpu`	(`I.i`) Number of CPUs
P	`cpu.context_switch`	(`R.f`) context switches rate
P	`cpu.interrupt`	(`R.f`) interrupts rate
P	`cpu.syscalls`	(`%.f`) sys time. (`R.f`) System call rate.
P	`cpu.mutex`	(`R.f`) Mutex rate
P	`cpu.forks`	(`R.f`) fork+vfork call rate
P	`cpu.execs`	(`R.f`) exec call rate
P	`cpu.swapout`	(`R.f`) number of swapouts per sec
P	`cpu.swapin`	(`R.f`) number of swapins per sec
P	`cpu.pgpgout`	(`R.f`) number of pages paged out per sec
P	`cpu.pgpgin`	(`R.f`) number of pages paged in per sec

Table A-1 Kernel Reader Data Hierarchy *(Continued)*

Node or Property?	Name of Node or Property	Value or Data Type and Brief Description
P	cpu.pgout	(R.f) number of page outs per sec
P	cpu.pgin	(R.f) number of page ins per sec
P	cpu.run_queue_length	((A,f) number of jobs waiting to be run
P	cpu.swap_queue_length	(R.f) number of jobs waiting on swap
P	cpu.io_queue_length	(R.f) number of jobs waiting for I/O
N	cpu.cpuX	Individual CPU data (X=0, 1, 2...)
P	cpu.cpuX.instance	(I.i) CPU instance (i.e. the X in the label)
P	cpu.cpuX.busy	(%.f) CPU busy time (usr+sys, 100.0 == 100%)
P	cpu.cpuX.user	(%.f) user time (100.00 == 100%)
P	cpu.cpuX.sys	(%.f) sys time (100.00 == 100%)
P	cpu.cpuX.wait	(%.f) wait time (100.00 == 100%)
P	cpu.cpuX.idle	(%.f) idle time (100.00 == 100%)
P	cpu.cpuX.context_switch	(R.f) context switches rate
P	cpu.cpuX.interrupt	(R.f) interrupts rate
P	cpu.cpuX.syscalls	(R.f) System call rate
P	cpu.cpuX.mutex	(R.f) Mutex rate
P	cpu.cpuX.forks	R.f) fork/vfork call rate
P	cpu.cpuX.execs	(R.f) exec call rate
P	cpu.cpuX.swapouts	(R.f) number of swapouts per sec
P	cpu.cpuX.swapins	(R.f) number of swapins per sec
P	cpu.cpuX.page_paged_out	(R.f) number of pages paged outs per sec
P	cpu.cpuX.page_paged_in	(R.f) number of pages paged ins per sec
P	cpu.cpuX.pageouts	(R.f) number of page out per sec

Table A-1 Kernel Reader Data Hierarchy *(Continued)*

Node or Property?	Name of Node or Property	Value or Data Type and Brief Description
P	cpu.cpuX.pageins	(R.f) number of page in per sec
N	mem	Physical memory related
P	mem.mem_avail	(I.f) total memory available (physmem) (MBytes)
P	mem.mem_inuse	(A.f) physical memory in use (MB)
P	mem.mem_inuse_p	(%.f) physical memory in use (%)
P	mem.mem_free	(A.f) physical memory free (MB)
P	mem.swap_avail	(A.f) swap space available (MB)
P	mem.swap_resv	(A.f) swap space reserved (MB)
P	mem.swap_alloc	(A.f) swap space allocated (MB)
P	mem.swap_free	A.f) swap space free (MB)
P	mem.swap_total	(I.f) total swap space (MB)
P	mem.swap_devfree	(A.f) total free swap space (MB)
N	disk	For disk I/O, summary on all disks
P	disk.ops	(R.f) avg. disk op.(r+w) rate (op/s)
P	disk.reads	(R.f) avg. disk read rate (op/s)
P	disk.write	(R.f) avg. disk write rate (op/s)
P	disk.nread	(R.f) Number of bytes read (bytes/s)
P	disk.nwritten	(R.f) Number of bytes written (bytes/s)
P	disk.queuelength	(A.f) avg disk queue length
P	disk.waittime	(A.i) avg Wait time (ms)
P	disk.runtime	(A.i) avg run time
N	disk.XXX	Individual disks (disk name as title, sd3)
P	disk.XXX.reads	(R.f) avg. disk read rate (op/s)
P	disk.XXX.write	(R.f) avg. disk write rate (op/s)
P	disk.XXX.nread	(R.f) number of bytes read (KB/s)
P	disk.XXX.nwritten	(R.f) number of bytes written (KB/s)

Table A-1 Kernel Reader Data Hierarchy *(Continued)*

Node or Property?	Name of Node or Property	Value or Data Type and Brief Description
P	disk.XXX.busy	(%.f) disk busy (100.0 = 100%)
P	disk.XXX.svctime	(A.i) Avg. service time (ms)
P	disk.XXX.waittime	(A.i) Avg. wait time (ms)
P	disk.XXX.runtime	(A.i) Avg. run time (ms)
P	disk.XXX.queuelength	(A.f) Avg. queue length
N	net	Network-related values, all interfaces
P	net.nnet	(I.i) # of net interface
P	net.oflo	(R.f) overflow error
P	net.uflo	(R.f) underflow error
P	net.crc	(R.f) CRC error
P	net.framming	(R.f) frame error
P	net.collisions	(R.f) collisions
P	net.oerrors	(R.f) output errors
P	net.ierrors	(R.f) input errors
P	net.opackets	(R.f) output packet count (pkt/s)
P	net.ipackets	(R.f) input packet count (pkt/s)
N	net.XXX	Each individual net interface, name as label (i.e. le0)
P	net.XXX.oflo	(R.f) overflow error
P	net.XXX.uflo	(R.f) underflow error
P	net.XXX.crc	(R.f) CRC error
P	net.XXX.framming	(R.f) frame error
P	net.XXX.collisions	(R.f) collisions
P	net.XXX.oerrors	(R.f) output errors
P	net.XXX.ierrors	(R.f) input errors
P	net.XXX.opackets	(R.f) output packet count (pkt/s)
P	net.XXX.ipackets	(R.f) input packet count (pkt/s)

Table A-1 Kernel Reader Data Hierarchy *(Continued)*

Node or Property?	Name of Node or Property	Value or Data Type and Brief Description
N	`process`	Process related information
N	`process.PID`	Individual process, PID as label (i.e. 123)
P	`process.PID.pid`	(`I.i`) PID as integer
P	`process.PID.ppid`	(`I.i`) parent PID as integer
P	`process.PID.uid`	(`I.i`) user ID
P	`process.PID.command`	(`I.s`) command line
P	`process.PID.tty`	(`I.s`) control tty
P	`process.PID.uname`	(`I.s`) user name
P	`process.PID.nice`	(`I.i`) process nice value
P	`process.PID.upcnt`	(`%.f`) CPU time in user mode (%), last rep. interval
P	`process.PID.spcnt`	(`%.f`) CPU time in system mode (%), last rep. interval
P	`process.PID.tpcnt`	(`%.f`) total CPU time (%), last rep. interval
P	`process.PID.utime`	(`I.s`) total CPU time in user mode (str.), life time
P	`process.PID.stime`	(`I.s`) total CPU time in system mode (str.), life time
P	`process.PID.ttime`	(`I.s`) total CPU time in total (str.), life time
P	`process.PID.start`	(`I.s`) process start up time, (str.)
P	`process.PID.deadtime`	(`I.s`) process (approx.) dead time, (str.)
P	`process.PID.syscall`	`I.s`) syscall name the process is in. (solaris)
P	`process.PID.sz`	(`A.f`) process size (MB)
P	`process.PID.rssz`	(`A.f`) resident set size (MB)
P	`process.PID.priority`	(`I.i`) priority
P	`Process.PID.status`	(`I.s`) Process status

Table A-1 Kernel Reader Data Hierarchy *(Continued)*

Node or Property?	Name of Node or Property	Value or Data Type and Brief Description
P	process.PID.ioch	(S.i) total IO characters (in interval)
P	process.PID.vcsw	(S.i) voluntary context switch
P	process.PID.invcsw	(S.i) involuntary context switch
P	process.PID.signals	(S.ii) signals received
P	process.PID.swaps	(S.i) times being swapped
P	process.PID.majf	(S.i) major page fault
P	process.PID.minf	(S.i) minor page fault
P	process.PID.receive	(S.i) receive system calls
P	process.PID.send	(S.i) send system calls
P	process.PID.bwrite	(S.i) block write calls
P	process.PID.bread	(S.i) block read calls
P	process.PID.nlwp	(S.i) number of light weight processes associated with the process

A

Config Reader B

This appendix describes the Config Reader. The Config Reader determines the configuration of a machine, along with the status of its different components. Its output is depicted in two forms:

- Physical view
- Logical view

This appendix provides a listing of the Config Reader data hierarchy in tabular form. Use the information in this appendix if you want to write event rules that monitor the output of the Config Reader. Note that node properties are always quoted. Where there is a default value for a particular node, the property name is followed by (*) and its value is listed.

Table B-2 uses these abbreviations:

- I = Integer
- H = Hexadecimal
- A = ASCI

Note – Names and types are subject to change and therefore do not constitute an interface definition.

B

Common Terms

These terms define information in Table B-2.

Table B-1 Definitions of Common Terms

Term	Definition
`P-instance`	Decimal or hexadecimal number acquired by reading `/etc/path_to_inst`, which maps physical device names to the instance number. Refer to the man page for more information.
`P-upa-portid`	Each board has two CPU slots, A and B, which are given unique numbers. For example, board 0 has upa-portid 0 and 1, and board 1 has upa-portid 2 and 3, etc. portid / 2 = slot number. This property exists for SBus and CPU units.
`P-upa-mid`	Always equal to upa-portid. This property is displayed for fhc, sbus, cpu-unit and central modes.
`P-hpu`	Hot plugable units can be replaced without powering the system down.

Config Reader Data Hierarchy

Table B-2 lists the Config Reader data hierarchy. As implied by the name, nodes connect to each other to form a tree that describes the configuration of the host. They can also be electrical entities. For example, the `system_3v node` is not a physical entity but is the input voltage to system boards.

Each node has a set of properties that describes its state; for example, if an LED is on of off, or if a voltage level is acceptable.

The first column of the table contains the data type of the property. Note the Config Reader always encloses the value of a property between double quotes (""). The second column contains the name of the node or property. An "N-"

prefix means it is a node and a "P-" prefix means it is a property. Each node is followed by its set of properties. The third column explains the meaning of the node or property.

Note – Node and property names and types are subject to change. This appendix does not constitute an interface definition.

Table B-2 Config Reader Data Hierarchy

Type or Default Value	Node	Description
	N-AC_PS	AC connector and switch
“A”	P-status	“OK”/”FAIL”
	N-SUNW,fas	Host Bus Adapter Driver
“I”	P-tape_count	Number of tapes attached to a particular slot
“I”	P-disk_count	Number of disks attached to a particular
“I”	P-instance	Refer to the section “Common Terms” at the beginning of this appendix
“A”	P-device_type*	“scsi”
	N-SUNW,ffb(N)	UPA color frame buffer and graphics accelerator. N= instance number
“A”	P-buffer	“Single”/”Double”
“A”	P-3DRAM	3-D RAM model and version number
“A”	P-DAC	Revision of DAC chip
“H”	P-FBC_version	
“A”	P-revision	Revision of ffb board

Table B-2 Config Reader Data Hierarchy *(Continued)*

Type or Default Value	Node	Description
"I"	P-instance	Refer to the section "Common Terms" at the beginning of this appendix
"I"	P-upa-portid	Refer to the section "Common Terms" at the beginning of this appendix.
"I"	P-width	Pixels
"I"	P-height	Pixels
"A"	P-device_type*	"display"
"A"	P-model	
	N-SUNW,hme	Fast Ethernet device driver
"I"	P-instance	Refer to the section "Common Terms" at the beginning of this appendix
"A"	P-device_type*	"network"
	N-SUNW,leo	Double-buffered SBus color frame buffer and graphics accelerator
"I"	P-instance	Refer to the section "Common Terms" at the beginning of this appendix.
"I"	P-width	Pixels
"I"	P-height	Pixels
"A"	P-model	
"A"	P-device_type*	"display"
	P-SUNW,pln(N)	Sparcstorage Array
"I"	N-disk_count	Number of attached disks
"A"	N-device_type	Sparcstorage Array
"I"	N-instance	Refer to the section "Common Terms" at the beginning of this appendix

Table B-2 Config Reader Data Hierarchy *(Continued)*

Type or Default Value	Node	Description
	N-SUNW,rtvc	Real-time video capture
"I"	P-instance	Refer to the section "Common Terms" at the beginning of this appendix
	P-model	
	P-SUNW,soc(N)	Serial Optical Controller device driver. N = instance number.
"A"	P-device_type*	"Serial Optical Channel Processor Host Adapter"
"I"	P-instance	Refer to the section "Common Terms" at the beginning of this appendix
	P-model	
	P-soc-fcode	
"I"	P-soc-wwn	World Wide Web device name
"I"	P-port-wwns	
	N-board(N)	N= board number
"yes"	P-hpu*	hot plug unit?
"A"	P-temperature	"N C", where N is an integer

Table B-2 Config Reader Data Hierarchy *(Continued)*

Type or Default Value	Node	Description
	`P-state=`	"active": Normal state for boards after the operating system has booted "disabled": Disabled by the PROM because it was listed in the 'disabled-board-list' property in the options node "failed": A board that is failed by POST and powered off in the testing phase of PROM operations "hot-plug": Intermediate state after a board is hotplugged "low-power": The state that hotplug boards are "unknown": Unused
"yes"	`P-fru*`	field replaceable unit?
"A"	`P-type`	One of: "clock", "cpu/memory", "Dual sbus I/O", "disk-board", "sbus/FFB I/O", "pci I/O".
"I"	`P-board-num`	For Ultra-Enterprise-3000: up to 4
"A"	`P-memory_size`	"N MB", N is an integer (CPU/memory boards only)
	`N-cdrom`	
"I"	`P-mounted_partitions`	
"A"	`P-device_type`	"CD-ROM"
"A"	`P-name`	For example"`c0t6d0`"
"I"	`P-instance`	Refer to the section "Common Terms" at the beginning of this appendix
	`N-central`	Only exists under the clock board

Table B-2 Config Reader Data Hierarchy *(Continued)*

Type or Default Value	Node	Description
“I”	`P-instance`	Refer to the section “Common Terms” at the beginning of this appendix
“I”	`P-upa-mid`	Refer to the section “Common Terms” at the beginning of this appendix
	`N-clock-board`	
“I”	`P-instance`	Refer to the section “Common Terms” at the beginning of this appendix
“A”	`N-cpu-unit(N)`	N is equal to unit, below
“yes”	`P-fru*`	Field replaceable unit?
“A	`P-clock-frequency`	“N MHz”, N is an integer
“A”	`P-cpu-type`	For example “sparc”
“A”	`P-status`	“online”/”offline”
“A”	`P-model`	
“I”	`P-processor-id`	2*board number+1 if cpu-unit== “B”
“A”	`P-unit`	“A” or “B” (each board takes two cpu’s)
“I”	`P-upa-mid`	Refer to the section “Common Terms” at the beginning of this appendix
“I”	`P-upa-portid`	Refer to the section “Common Terms” at the beginning of this appendix
“A”	`P-icache-size`	“N KB” or “N MB”, where N is a fixed point number
“A”	`P-dcache-size`	“N KB” or “N MB”, where N is a fixed point number

Table B-2 Config Reader Data Hierarchy *(Continued)*

Type or Default Value	Node	Description
"A"	P-ecache-size	"N KB" or "N MB", where N is a fixed point number
"A"	P-device_type*	"cpu"
"A"	P-board#	Board number on which this node resides
	N-disk_fan	Exists only in Ultra Enterprise 3000
"yes"	P-hpu*	hotplug unit?
"A"	P-status	"OK"/"FAIL"
	N-fan	In the Ultra-Enterprise 4000,5000 and 6000, the AC_PS has a fan node
	p-status	"OK"/"FAIL"
	N-fhc	Fire Hose Controller. There is one associated with each AC. Contains board CSR, controls on-board devices such as PROM, RAM and UARTS
"I"	P-instance	Refer to the section "Common Terms" at the beginning of this appendix
"I"	P-upa-mid	Refer to the section "Common Terms" at the beginning of this appendix
"I"	P-version#	Version number
"I"	P-board#"	Board number on which this node resides
"A"	P-model	
	N-ac	Address controller

Table B-2 Config Reader Data Hierarchy *(Continued)*

Type or Default Value	Node	Description
"I"	P-instance	Refer to the section "Common Terms" at the beginning of this appendix
"I"	P-version#	
"A"	P-model	
"A"	P-device_type*	"memory-controller"
	N-eeprom	Electrically programmable read-only memory
"A"	P-model	
	N-environment	
"I"	P-instance	Refer to the section "Common Terms" at the beginning of this appendix
	N-flashprom	
"I"	P-instance	Refer to the section "Common Terms" at the beginning of this appendix
"A"	P-model	
	N-sbus-speed	
The following ten nodes are not physical entities. The following data is the output of different power supplies in the system used for hotplugging, peripherals, and so on.		
	N-hot_plug_charges	
"no"	P-fru*	field replaceable unit?
	N-auxiliary_5v	
"A"	P-status	"OK"/"FAIL"
"no"	P-fru*	field replaceable unit?
	N-peripheral_12v	

Table B-2 Config Reader Data Hierarchy *(Continued)*

Type or Default Value	Node	Description
"A"	P-status	"OK"/"FAIL"
"no"	P-fru*	field replaceable unit?
	N-peripheral_12v_precharge	
"A"	P-status	"OK"/"FAIL"
"no"	P-fru*	field replaceable unit?
	N-peripheral_5v	
"A"	P-status	"OK"/"FAIL"
"no"	P-fru*	field replaceable unit?
	N-peripheral_5vprecharge	
"A"	P-status	"OK"/"FAIL"
"no"	P-fru*	field replaceable unit?
	N-system_5v	
"A"	P-status	"OK"/"FAIL"
"no"	P-fru*	field replaceable unit?
	N-system_5v_ precharge	
"A"	P-status	"OK"/"FAIL"
"no"	P-fru*	field replaceable unit?
	N-system_3v	
"A"	P-status	"OK"/"FAIL"
"no"	P-fru*	field replaceable unit?
	N-system_3v_precharge	
"A"	P-status	"OK"/"FAIL"
"no"	P-fru*	field replaceable unit?
	N-keyswitch	
"A"	P-position	"Normal Mode," "Secure mode," or "Diagnostic mode"

Table B-2 Config Reader Data Hierarchy *(Continued)*

Type or Default Value	Node	Description
	N-led(N),	Power supplies have 2 LEDs: green (off = no power) and amber (on=FAIL). Boards have 3 LEDs. Refer to Maintenance Manual for meanings of led combinations.
"I"	P-led-num	1, 2, or 3
"A"	P-state	"on"/"off"
	N-peripheral_PS(N)	184w power supply for SCSI peripherals, additional power for boards in system and pre-charge power for hotplugging boards.
"yes"	P-hpu*	hot plug?
"A"	P-status	"OK"/"FAIL"
"I"	P-unit#	Unique number of this power supply
"yes"	P-fru*	field replaceable unit?
	N-nf(N)	SBus FDDI Driver
"A"	ether	Ethernet address. For example "8:0:20:75:af:f8"
"I"	inet	Internet address. For example "129.146.65.45"
"A"	symbol	Network name corresponding to the inet address above
"A"	name	Driver name. For example "nf0"
"I"	instance	Refer to the section "Common Terms" at the beginning of this appendix
"A"	model	
"A"	device_type	"network"
	N-power_supply(N)	300w board supply, N=unit number below

Table B-2 Config Reader Data Hierarchy *(Continued)*

Type or Default Value	Node	Description
"yes"	P-hpu*	hot plug?
"A"	P-status	"OK" / "FAIL"
"I"	P-unit#	Unique number of this power supply
"yes"	P-fru*	field replaceable unit?
	N-sbus(N)	Peripheral bus, N=instance number.
"I"	P-instance	Refer to the section "Common Terms" at the beginning of this appendix
"I"	P-upa-portid	Refer to the section "Common Terms" at the beginning of this appendix
"I"	P-upa-mid	Refer to the section "Common Terms" at the beginning of this appendix
	P-version#	Refer to the section "Common Terms" at the beginning of this appendix
"A"	P-board#	Refer to the section "Common Terms" at the beginning of this appendix
"A"	P-device_type*	
	P-model	Board number on which this node resides
"I"	P-network_count	"sbus"
	N-remote_console	"A"
	P-status="Enabled"	Network devices attached to this SBus
	P-sd(0)	

Table B-2 Config Reader Data Hierarchy *(Continued)*

Type or Default Value	Node	Description
"yes"	P-fru*	
"I"	P-mounted_partitions	
"A"	P-device_type*	field replaceable unit?
"A"	P-name	
"I"	P-instance	"disk" For example, "c0t0d0"
The following data is the output of Config Reader if the disk has mounted partition.		
"A"	P-name	For example "c0t0d0s0"
"I"	P-percent_used	For example "49%"
"I"	P-total_bytes	"n mbytes" or "n kbytes"
"I"	P-avail_bytes	"n mbytes" or "n kbytes"
"I"	P-used_bytes	"n mbytes" or "n kbytes""
"A"	P-device_type*	"partition"
"A"	P-export name	For example "c0t0d0s7"
	N-simm(0)	
"A"	P-board_reference_number	
"I"	P-slot	CPU/memory slot where this SIMM resides
"A"	P-size	"N MB", where N is an integer
"A"	P-type	"dram"
"yes"	P-fru*	field replaceable unit?
	N-sram	Static RAM
"I"	P-instance	Refer to the section "Common Terms" at the beginning of this appendix.
	N-st(N)	SCSI tape, N is the instance number

Table B-2 Config Reader Data Hierarchy *(Continued)*

Type or Default Value	Node	Description
"yes"	`P-fru*`	field replaceable unit?
"A"	`model`	For example "Exabyte EXB-8500 8mm"
"A"	`name`	For example "`/dev/rmt/0n`"
"A"	`status`	"ok"
"A"	`device_type`	"tape drive"
"I"	`instance`	"4"
	`N-system`	Contains general properties of the system
"I"	`P-total_processors`	Number of processors in the machine
"A"	`P-total_memory`	"N MB", where N is an integer
"I"	`P-total_disks`	
"I"	`P-total_tape_devices`	
"I"	`P-sample number`	since daemon started on server
"A"	`P-timestamp`	For example `Tue Mar 26 10:33:08 1996`
"A"	`P-hostname`	Machine name
"A"	`P-OS`	Value as returned by `uname`(2) in release field
"A"	`P-OS_version`	Value as returned by `uname`(2) in version field
"A"	`P-architecture`	For example, `sparc, mc68030, m32100, or 8386`
"A"	`P-machine`	For example, `sun4u`
"A"	`P-platform`	Specific model of the hardware platform.
"I"	`P-serial_number`	
"A"	`P-System_clock_frequency`	"N MHz", where N is an integer.

Glossary

action

Tells Solstice SyMON what to do when a rule is active. Actions can be used to notify users of a situation that may require attention. An action might be to create an event, to send a record of the event to the Event Log to generate an email message about the event, to use a modem to dial a beeper, or to execute a program. You may define new actions as needed.

agent

The program or daemon that collects data from the server and provides it to the rest of Solstice SyMON.

alarm

A predefined procedure for event rules that makes an event active, creates a log entry in the Event Log for the event, optionally posts a message to the Event Viewer, and highlights the affected node in the appropriate console.

attributes

Elements of an event rule, including the number of the rule, condition, actions, level, priority, and severity.

client

In the client-server model, the client is a machine that remotely accesses resources of a server. The server provides a service such as greater compute power or access to a large database, or larger memory capacity.

complex rule

An event rule in which a single rule can generate multiple events and log entries by searching a range of parameters. Complex rules make the Event Handler evaluate the condition for multiple components or systems. For example, complex rules can be used to make the Event Handler check the condition for each board on a system. The `MULTI` parameter must be used to designate a complex rule.

condition

A Tcl string that defines when a rule is active. This is the only mandatory part of a rule. A condition does not have a tag. Examples include a failed board or a DNLC long name ratio greater than 25 percent.

end_alarm

A predefined procedure for event rules that closes the log entry in the Event Log for the event, deletes the event from the Event Viewer, and removes any console.

event

An action that is used to notify other applications of a condition, typically hardware or operating system conditions, that may require the attention of the system administrator.

Event Handler

A subsystem of Solstice SyMON that reports events based on a set of supplied rules. The Event Handler collects data from monitoring agents and evaluates them for conditions defined within its rules. When a condition is true, an event occurs and the Event Handler implements any specified action(s) in the rule. When the condition that generated the event no longer exists, the Event Handler may run a special action such as closing the event.

event log

A log in which the event generator records events. It is an ASCII file with fixed length records.

Event Viewer

A console within the Solstice SyMON GUI that displays the Event Log.

EVENTS variable

A list of event rules specified in the file, `/etc/opt/SUNWsymon/rules.tcl`.

graphical user interface (GUI)

The Solstice SyMON user interface is a CDE/Motif-based application that receives and displays data from all of the agents of Solstice SyMON.

Kernel Data Catalog

Displays a hierarchical schematic view of the performance parameters that Solstice SyMON monitors.

Launcher

The main window of Solstice SyMON. Each button on the console launches one of the seven Solstice SyMON consoles: Event Viewer, Log Viewer, Physical View, Logical View, Kernel Data Catalog, Process Viewer, and the On-Line Diagnostics Screen. It also reports on and controls overall system functions (for example, its agents.)

Log Viewer

Displays entries that appear in the system log file.

Logical View

Displays a hierarchical schematic view of system components.

MULTI

A special event rule attribute used to permit the rule to check a list of node variables for multiple events off a single rule. For example, the `MULTI` string can be used to make the Event Handler check a condition of a rule for each board in the system.

ON_OPEN

An event rule attribute that tells Solstice SyMON what to do when the condition for a rule becomes true.

ON_CLOSE

An event rule attribute that is interpreted when a rule is no longer true.

ON_CONTINUE

An event rule attribute that is interpreted when a rule continues to be true two or more times in a row.

ON_SHUTDOWN

An event rule attribute that is interpreted when the system shuts down.

Physical View

Displays pictorial images of the server. The images are updated dynamically as displayed components change state.

PRIORITY

A rule attribute. A priority for the rule for the user to interpret that is used in conjunction with SEVERITY.

procedures file

Defines procedures used by Solstice SyMON rules. The file must exist, but can be empty. The file is located in `/etc/opt/SUNWsymon/event_gen.tcl`).

Process Viewer

Displays information on processes running on the server. Information includes time the process was started and by whom, the total CPU that the process has used, and its current status.

RULE

Event rule attribute that indicates the rule number.

rule

A condition and a set of actions associated with that condition. Rules are always in the EVENTS Tcl variable, which is located in the file, `/etc/opt/SUNWsymon/rules.tcl`.

rules functions

A rules function is a predefined procedure such as `ON_OPEN` or `ON_CLOSE;` it is used within an event rule, such as `alarm` and `end_alarm`.

server

In the client-server model, the server is a machine with compute resources or large data storage capacity. Client machines can remotely access and use these resources.

SEVERITY

An event rule attribute that indicates the severity of the rule for the user to interpret. Used in conjunction with PRIORITY.

simple rule

An event rule that has a one-to-one correspondence between the rule and an event. A simple rule does not check a wildcard list of variables. As a consequence, it can only generate a single event.

System Meter

A GUI window within the Kernel Data Catalog that displays graphs of monitored performance parameters.

tag

A tag is a textual string associated with a user-defined item.

LABEL_DATA

Format for attributes in Solstice SyMON rules. A tag is a textual string associated with a value. Valid tags are `RULE`, `ON-OPEN`, `ON-CLOSE`, `ON-CONTINUE`, `ON-SHUTDOWN`, `PARAMETERS`, `MULTI`, `SEVERITY`, and `PRIORITY`.

Tcl

A scripting language used to define Solstice SyMON rules.

Index

F

G

H

I

K

L

M

N

O

P

R

S

T

V

W