Tru64 UNIX

Managing Online Addition and Removal

Part Number: AA-RPUFA-TE

June 2001

Product Version: Tru64 UNIX, Version 5.1A

This manual discusses Online Addition and Removal of system components and related topics.

Compaq Computer Corporation
Houston, Texas

Contents

4 Online Addition and Removal

5 Service Tools

6 Memory Trolling

Index

Examples

Figures

Tables

About This Manual

This manual provides guidelines and management and configuration techniques used for Online Addition and Removal (OLAR) of system components using the Tru64™ UNIX operating system software. You will be able to accomplish these tasks on any system that has OLAR capabilities, using the System Management applications: SysMan Station, SysMan Menu, and the `hwmgr` command. Many of the tasks discussed in this book can be done on systems without OLAR capabilities as well. Related topics including component indictment and automatic deallocation as well as related service tools also are discussed.

Audience

This manual is intended for system administrators, service technicians and system operators who are responsible for managing and configuring a Tru64 UNIX operating system. This manual is not platform specific because a subset of the features discussed are available on all systems. Specifically, you may remove, add, or replace CPUs only on platforms that support CPU OLAR. For Tru64 UNIX Version 5.1A, this is specifically the AlphaServer™ GS160 and GS320 systems.

System administrators, service technicians and system operators should have extensive knowledge of their applications and hardware configurations prior to initiating an OLAR operation. Users should have extensive knowledge of operating system concepts, commands, and utilities.

If you plan to use your system in a clustered computing environment, see the TruCluster™ Server documentation.

Features

The operating system offers various System Management applications that support Online Addition and Removal. The tools that support OLAR operations are SysMan Station, SysMan Menu, and the `hwmgr` command.

The following list summarizes the features offered in this release of the operating system:

Central Processor Unit (CPU) OLAR	Allows you to perform capacity expansion, CPU upgrades, and replace failed CPUs, without having to shut down the entire system or an operating instance. On

	systems with this capability, the CPUs are separately powerable. CPU status also can be changed to off line, on line.
Component Indictment	Provides proactive error notification of potentially failing system components. Posts events through the EVM Event Manager to notify all interested applications.
Automatic Deallocation	Allows you to decide whether to deallocate components without user intervention when a component is indicted.
Service Applications	A set of service applications for performing system diagnostics, system management and configuration.
Memory Troller	Systematically reads and writes all of the available physical memory in an effort to locate and correct single-bit memory errors.

Organization

This manual is organized as follows:

Chapter 1	Provides an introduction to Online Addition and Removal and related features.
Chapter 2	Describes concepts and features important to increase system availability. References are made to other existing documentation where needed.
Chapter 3	Describes the Component Indictment facility that informs operators that a component has been identified as a potential point of failure and the Automatic Deallocation policy settings that decide what to do when this situation arises.
Chapter 4	Describes how to perform Online Addition and Removal (OLAR) operations with the various System Management applications and the command line utilities.

Chapter 5	Introduces the user to the Compaq service applications and describes how best to use the service applications to monitor your system and perform system diagnostics.
Chapter 6	Discusses the Memory Trolling capabilites of the operating system.

Related Documentation

The following documents are useful references when you are configuring hardware, and performing system management and configuration tasks:

- *System Administration* Manual

  This manual describes how to configure, use, and maintain the operating system. It includes information on general day-to-day activities and tasks, changing system configurations, and locating and eliminating sources of trouble. This manual is intended for the system administrators responsible for managing the operating system. It assumes a knowledge of operating system concepts, commands, and configurations.

- *Network Administration: Connections* Manual

  This manual describes how to configure and manage the network interfaces and network transports, and solve problems that might arise on systems running the Tru64 UNIX operating system software.

- *Technical Overview* Manual

  This manual describes the major components of the operating system. This manual also describes enhancements made to the operating system and provides detailed information on various aspects of the operating system.

- *System Configuration and Tuning* Manual

  This manual describes how to plan, set up, and tune high-performance and high-availability systems running the operating system.

- TruCluster Server *Cluster Technical Overview*

  This manual describes the major components and features of the TruCluster Server product.

- TruCluster Server *Cluster Hardware Configuration* Manual

  This manual describes how to set up and maintain the hardware configuration for a cluster server.

- Compaq AlphaServer GS320 technical resources web site:
  `http://www.compaq.com/alphaserver/gs320/gs320_tech.html`

- Compaq Documentation for the Compaq Continuous Profiling Infrastructure (DCPI) applications:
  http://www.tru64unix.compaq.com/dcpi/documentation.htm
- Compaq Documentation for the WEBES (Web-Based Enterprise Services) suite of tools:
  http://www.support.compaq.com/svctools/webes/index.html

The Tru64 UNIX documentation is available on the World Wide Web at the following URL:
http://www.tru64unix.compaq.com/docs

Icons on Tru64 UNIX Printed Manuals

The printed version of the Tru64 UNIX documentation uses letter icons on the spines of the manuals to help specific audiences quickly find the manuals that meet their needs. (You can order the printed documentation from Compaq.) The following list describes this convention:

G	Manuals for general users
S	Manuals for system and network administrators
P	Manuals for programmers
R	Manuals for reference page users

Some manuals in the documentation help meet the needs of several audiences. For example, the information in some system manuals is also used by programmers. Keep this in mind when searching for information on specific topics.

The *Documentation Overview* provides information on all of the manuals in the Tru64 UNIX documentation set.

Reader's Comments

Compaq welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: `readers_comment@zk3.dec.com`

  A Reader's Comment form is located on your system in the following location:

  `/usr/doc/readers_comment.txt`

Please include the following information along with your comments:

- The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate Compaq technical support office. Information provided with the software media explains how to send problem reports to Compaq.

Conventions

The following conventions are used in this manual:

`% cat`	Boldface type in interactive examples indicates typed user input.
Colored ink	Colored ink indicates information that you enter from the keyboard or a screen object that you must choose or click on.
`cat(1)`	A cross-reference to a reference page includes the appropriate section number in parentheses. For example, `cat(1)` indicates that you can find information on the `cat` command in Section 1 of the reference pages.
Return	In an example, a key name enclosed in a box indicates that you press that key.
#	A number sign represents the superuser prompt.
[|] { | }	In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.

`%` `$`	A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells.
`file`	Italic (slanted) type indicates variable values, placeholders, and function argument names.
⋮	A vertical ellipsis indicates that a portion of an example that would normally be present is not shown.

1

Introduction to Online Addition and Removal

This chapter introduces the following topics:

- Component Indictment and Automatic Deallocation (Section 1.2)
- Online Addition and Removal (Section 1.3)
- Service Tools (Section 1.4)
- Memory Troller (Section 1.5)

1.1 Overview

This chapter introduces Online Addition and Removal (OLAR) of components and the features that interact with it in order to increase a system's availability. These features help you maintain system availability and availability of services by addressing the following factors:

- Minimizing scheduled and unscheduled down time for capacity expansion and component upgrades
- Recovering from failures
- Proactively identifying potential failures

The topics in this chapter can help minimize impact to a system's availability by addressing fault detection (Compaq Analyze), fault anticipation and avoidance (component indictment, automatic deallocation, and memory troller), and recovery (Online Addition and Removal).

1.2 Component Indictment and Automatic Deallocation

Component indictment is a proactive error notification from a fault analysis utility, indicating that a component is experiencing high incidence of correctable errors, and therefore should be serviced. Component indictment involves the process of analyzing specific failure patterns from error log entries, either immediately or over a given time interval, and recommending a component's removal. The fault analysis utility signals the operating system that a given component is suspect. This causes the operating system to distribute the fault information through an indictment event. Interested applications, including SysMan Station, and the Automatic Deallocation

Facility can update their state information, and take appropriate action if so configured.

The Automatic Deallocation facility of the operating system can be configured by the system administrator to automatically put off line an indicted component.

For more information on Component Indictment and Automatic Deallocation, see Chapter 3.

1.3 Online Addition and Removal

Online Addition and Removal is the ability to add or remove critical system components while the operating system services and applications continue to run.

Online Addition and Removal management is used to expand capacity, upgrade components, and replace failed components without adversely affecting the availability of the system. This functionality, sometimes referred to as hot-swap, provides the benefits of increased system up time and availability during both scheduled and unscheduled maintenance. Starting with Tru64 UNIX V5.1A, CPU OLAR is supported.

For more information on Online Addition and Removal, see Chapter 4.

1.4 Service Tools

Applications in the WEBES suite of tools provide a core of common service tool functionality, including hardware diagnosis, operating system analysis, system configuration and revision reporting capabilities. These tools have been integrated into the SysMan suite of tools in order to allow easy, centralized access. For more information, see Chapter 5. Additionally, the Compaq Analyze component of the WEBES suite is used in the Component Indictment process.

1.5 Memory Troller

The memory troller is an operating system mechanism that proactively locates and scrubs correctable memory errors. The memory troller systematically reads each memory location at a configurable rate. If it discovers a correctable memory error, it triggers the just-in-time scrubbing mechanism. For more information on memory trolling, see Chapter 6.

2

Availability Considerations

This chapter describes some high level considerations regarding configuration of your system and its components for increased availability. These considerations pertain to either standalone systems or members of a cluster. This chapter does not intend to provide a comprehensive discussion of the considerations.

This chapter discusses the following topics:

- Redundancy of Systems and Components (Section 2.2)
- Configuring I/O for Availability (Section 2.3)

2.1 Overview

Proper system setup and configuration of your hardware is essential in maintaining its availability. You need to be aware of environmental factors, for example, room temperature, altitude and humidity. For the specific environmental specifications, see the User's Guide that came with your system. You also must ensure that you have access to the correct power requirements for your system.

The following sections provide descriptions and guidelines, so that you can ensure your system is configured for maximum system availability.

2.2 Redundancy of Systems and Components for Availability

One common method of assuring the availability of systems is redundancy. Redundancy of components, such as multiple CPUs or multipath I/O connections, is important to assuring operation that is as continuous as possible.

As an example, if only one component supplies a critical function, a failure in this component would stop a system from providing its services. This is known as a single point of failure. Redundancy of components allows the failure of a particular component to occur, yet not be a critical failure to the system or the services it provides.

Redundancy is handled by one of two methods: failover or multipathing.

When hardware components are configured for failover, a backup component on standby takes over for the failed component. This incurs the added cost of the second component, with no added performance, but is a simple and effective way to avoid the effects of a component failure. An example of this is NetRAIN network arrays, discussed in Section 2.3.1.

With multipathing, multiple components provide multiple paths for data to flow. This has the added benefit, in many cases, of increasing the performance of the system. If one of the components fails, the system continues functioning in a degraded state until repairs are made. An example of this is the Multipathing of SCSI or Fibre Channel, discussed in Section 2.3.3.

With either method, it is important to replace the failed component quickly in order to guarantee the system's ability to maintain availability.

Components that are capable of being replaced while the system remains on line allows servicing of your system without a loss of availability. Components that are capable of Online Addition and Removal (OLAR) are discussed in Chapter 4.

Systems that require absolute minimal down time of a system and its services may require redundancy of systems supplied by clustering. Clustering also has the benefit of easily allowing redundancy and failover of software using the Cluster Alias and Cluster Application Availability subsystems. For more information on clustering technology, see the TruCluster Server *Cluster Technical Overview*.

2.3 Configuring I/O

The following sections discuss configuration techniques for increasing the availability of your I/O devices, including references to documentation providing detailed configuration steps.

2.3.1 Using Redundant Array of Independent Network (NetRAIN) for Available I/O

NetRAIN (Redundant Array of Independent Network adapters) detects the physical loss of network connectivity and automatically switches traffic to a working network interface.

One network interface in the array of adapters is always active while the others remain idle. If the active interface fails, one of the idle set members comes on line with the same IP address.

For more information on NetRAIN configuration, see the *Network Administration: Connections* manual.

2.3.2 Using Link Aggregation for Available I/O

Link aggregation, or trunking, enables administrators to combine two or more physical Ethernet Network Interface Cards (NICs) and create a single logical link. (Upper-layer software sees this link aggregation group as a single logical interface.) The single logical link can carry traffic at higher data rates than a single interface because the traffic is distributed across all of the physical ports that make up the link aggregation group.

If one network interface in the aggregattion group fails, the remaining interfaces continue to provide connectivity with degraded bandwidth.

For more information on link aggregation see `lag(7)` or *Network Administration: Connections*.

2.3.3 Configuring SCSI and Fibre Channel for Multipath Redundancy

Multipath redundancy is the ability to connect more than one adapter to the same storage. The system automatically (in almost all cases) determines that the same storage sets are connected through multiple adapters and coordinates the access appropriately.

Multipath redundancy can be used to increase availability and increase performance (however, some configurations may experience a decrease in performance). Some configurations eliminate the single point of failure of the SCSI bus, while other configurations still will retain that single point of failure (the single SCSI bus that connects it all together). Multipath configurations can contain paths to storage that use either single busses or multiple busses.

Multibus is similar to multipath, and often confused with multipath. Multipath is the generic term used to refer to multiple adapters connected to the same storage. Multibus is a more specific term that refers to the capability of those devices to connect to multiple independent busses (or multiple ports).

Multibus configurations do not have the bus as a single-point of failure for I/O, while multipath using a single bus does.

For further discussion of hardware configurations for clusters and single systems, see *Cluster Hardware Configuration* or **http://www.tru64unix.compaq.com/docs/updates/TCR51_FC/TITLE.HTM**.

2.3.4 Configuring PCI Drawers

Multiprocessor systems such as a GS80/GS160/GS320 have multiple PCI drawers, each with its own power supply and connection to the system. To

avoid losing access to a service provided by a PCI card due to a failure of one of the drawers, you can increase your system's resilience by configuring PCI cards so redundant cards are in separate PCI drawers.

For example, if access to a network is supplied by two network cards in a NetRAIN set, placing one of the cards in one PCI drawer and the other in another PCI drawer will guard against failure of one of the drawers. If you were to place both cards in one PCI drawer, even if on separate busses in that drawer, you then would have a single point of failure that could remove the whole NetRAIN set and therefore remove access to the corresponding network.

2.3.5 Using AdvFS and LSM for I/O Availability

With AdvFS you can modify your storage configuration at any time without taking down the system. As your system requirements change, AdvFS allows you to easily adjust your storage size up or down to meet your requirements.

AdvFS also minimizes down time at reboots because it can have a faster boot time compared to UFS file systems because the file system does not need to be analyzed by the `fsck` command before boot.

AdvFS can incorporate Logical Storage Manager (LSM) volumes into the file system structure. AdvFS configured with LSM improves file system reliability and availability because AdvFS can take advantage of LSM features.

The Logical Storage Manager (LSM) software is an optional integrated, host-based disk storage management application. LSM uses Redundant Arrays of Independent Disks (RAID) technology to enable you to configure storage devices into a virtual pool of storage to protect against data loss, maximize disk use, improve performance, provide high data availability, and manage storage without disrupting users or applications accessing data on those disks.

LSM allows you to manage all of your storage devices, such as disks, partitions, or RAID sets, as a flexible pool of storage from which you create LSM volumes. You configure new file systems, databases, and applications, or encapsulate existing ones to use an LSM volume instead of a disk partition.

For more information, see *Logical Storage Manager*.

2.3.6 Choosing Hardware or Software RAID for I/O Availability

RAID can be used to increase the availability of storage. RAID also can benefit performance of storage access. The choice of whether to use hardware

or software RAID and which RAID level to implement are subject to your needed cost, performance and levels of availability.

Levels of RAID I/O

The following are the most common levels of RAID and a summary of their different capabilities:

RAID Level 0 supplies striping of data across multiple disks. This does not provide an increase in availability, but increases performance. RAID Level 0 often is combined with RAID Level 1.

RAID Level 1 supplies mirroring of data across disks. If one disk fails, all data is available because of the mirroring. RAID Level 1 increases availability. Increased costs arise due to duplication of storage. Write performance is also somewhat lowered when using RAID Level 1.

RAID Level 5 supplies striping across disks with stored parity data. If one of the hard drives fails, data still can be accessed at somewhat degraded performance until the failed disk is replaced and the RAID set is rebuilt. RAID Level 5 can be implemented in software, like LSM, but the overhead of parity checking usually calls for a hardware controller to achieve reasonable performance.

For more information, see *Logical Storage Manager*.

3

Component Indictment and Automatic Deallocation

This chapter discusses the Component Indictment and Automatic Deallocation facilities. Component indictment identifies system components that have a likelihood of future (potentially serious) failure based on a history of correctable non-fatal errors. This is done by analyzing specific failure patterns either immediately or over an extended period of time. The Automatic Deallocation facility provides the ability to automatically take an indicted component out of service.

This chapter discusses the following topics:

- Indictment of CPUs and memory pages (Section 3.1)
- Automatic Deallocation of CPUs and memory pages (Section 3.2)

3.1 Component Indictment

Component indictment is a proactive error notification from a fault-analysis utility. The component indictment process is intended to identify components that are incurring high or abnormal incidence of correctable errors, so that these components can be removed or repaired prior to them potentially causing a system panic.

The following are requirements for component indictment support:

- AlphaServer GS80/GS160/GS320
- Compaq Analyze V4.0 (included as part of the Web-Based Enterprise Services V4.0 product)
- A properly initialized binlog (`/var/adm/binary.errlog`) file, see `binlogd`(8)

The binlog error log must be maintained correctly for component indictment to function. The correct procedure for cleaning the binlog file is documented in `binlogd`(8). If you simply move the error log file and create a new file using `touch` without following the correct procedure, component indictment will not work as expected.

An external analysis program (currently Compaq Analyze) can notify the operating system when a component has encountered enough correctable errors to indicate that the component may fail soon. Upon receipt of the

indictment notification, the operating system posts an indictment event using the Event Management (EVM) subsystem. Administrators should investigate the source of any reported indictments and replace the indicted components as appropriate based on collaborative discussion with their service provider. Compaq Analyze currently supports indictments for CPUs and memory locations. Compaq Analyze supports EV6 and later processors.

Because indictment notification is posted to the Event Management (EVM) subsystem, any and all interested applications may subscribe to indictment events and take appropriate action. The Automatic Deallocation facility is one such application, which subscribes to indictment events and can be used to perform automatic deallocation of such indicted components. It also allows for execution of user-defined scripts, as discussed in Section 3.2.2 or Section 3.2.1 at the time of automatic deallocation. This avoids the need for an administrator to separately subscribe to these indictment events in order to handle them unless very specific processing is needed.

Additionally, if Compaq Analyze indicts a CPU, an immediate service call typically will be made to Compaq Services to allow the expedient scheduling of repair and replacement if a service obligation is in effect. For more information about Compaq Analyze, see Section 5.2. In addition to Compaq Analyze's features to contact your service representative, you also may set up pager or e-mail notification of component indictments based on EVM events using the EVM forwarding facility. See `evmlogger`(8) and `evmlogger.conf`(4) for more information.

If Compaq Analyze is unable to indict a specific component with certainty, but errors in a hardware subsystem are evident, there may be multiple indictments for a single failure source.

Every indictment event contains an urgency and probability value. The `probability` event variable will have one of up to three associated probabilities: high (100), medium (50), or low (1). For more information, see Section 3.1.4.

The `urgency` event variable identifies the seriousness of the problem.

If an indicted component is not placed off line within a 24 hour period, and correctable errors continue to be detected, another indictment may be issued by Compaq Analyze and another indictment event is posted if the urgency or probability of the indictment has changed.

The indicted state is persistent across system reboots and system initialization.

3.1.1 Indictment Process Overview

The process of component indictment follows this order:

1. A component such as a CPU or a memory location begins exhibiting correctable errors. These errors are written to the binary error log.
2. The fault analysis utility (Compaq Analyze) is notified automatically of each binary error log entry, reads the errors written to the binary error log, and performs an analysis of them. If the analysis concludes that the component potentially may have an unrecoverable error, the analysis program informs the operating system that the component should be considered for replacement, by issuing an indictment notification.
3. When the operating system receives an indictment, it sets the component's indictment attributes in the kernel and posts an indictment event using EVM. For an example indictment event, see Example 3–1.
4. The Automatic Deallocation facility listens to the indictment events and performs the appropriate deallocation dictated by the user-defined policy settings. This may include automatically putting off line a component or marking a memory page as bad.

   The SysMan Station also listens to these events and updates its display with the state of the system components. For information on SysMan Station, see Section 4.7.

3.1.2 Indictment Events

As a result of receiving an indictment notification from Compaq Analyze, the operating system posts an indictment event to the Event Management Subsystem (EVM). System Management applications subscribe to these indictment events. The SysMan Station (SMS) subscribes to indictment events so that it can change the indicted component's icon to show that the component is experiencing problems. An indictment event will cause a change in the status light for the System attention group in the SysMan Station Monitor View. For details on viewing indictment events, see Section 4.7. The Automatic Deallocation utility also subscribes to indictment events so that it can determine if automatic deallocation is required based on user-defined policy.

All indictment events have a prefix of `sys.unix.hw.state_change.indicted`. An example event for a CPU, which has a hardware ID (HWID) of 59, being indicted with a probablity of high, follows:

```
sys.unix.hw.state_change.indicted.high.cpu._hwid.59._hwcomponent.CPU4
```

Indictment events can be viewed at the command line using typical EVM methods. An example to view these events as they are posted would be:

```
# evmwatch -f '[name sys.unix.hw.state_change.indicted]' | evmshow
```

An example to view the events in the EVM event log would be:

```
# evmget -f '[name sys.unix.hw.state_change.indicted]' | evmshow
```

See EVM(5) or the *System Administration* manual for more information.

An example of a fully formatted event follows, using the command:

```
# evmget -f '[name sys.unix.hw.state_change.indicted]' | evmshow -D
```

Example 3–1 shows an indictment event for a CPU. It is indicted with a high probability. Example 3–2 shows an indictment event with a medium probability initiated concurrently with the CPU indictment.

Example 3–1: CPU Indictment Event

```
.
.
Formatted Message:
    Component State Change: Component "CPU0" has been indicted with a 'high'
    probability of fault (HWID=2, FRUID=11529776898687173375)

Event Data Items:
    Event Name        :
sys.unix.hw.state_change.indicted.high.cpu._hwid.2._hwc
                        omponent.CPU0
    Cluster Event     : True
    Priority          : 500
    PID               : 524288
    PPID              : 0
    Event Id          : 957
    Member Id         : 1
    Timestamp         : 08-May-2001 15:47:08
    Host IP address   : 16.69.242.74
    Host Name         : wild-one
    Cluster Name      : wild-bunch
    Format            : Component State Change: Component
"$_hwcomponent" has
                        been indicted with a 'high' probability of fault
                        (HWID=$_hwid, FRUID=$module_id)
    Reference         : cat:evmexp.cat:800

Variable Items:
    current_state (STRING) = "indicted"
    category (STRING) = "cpu"
    urgency (INT32) = 8
    probability (INT32) = 100
    total_indictments (INT32) = 2
    description (STRING) =
            "Excessive Correctable Memory Istream/Dstream Errors
reported by
             CPU0, CPU Slot0 in SoftQBB0 (HardQBB0)"
    initiator (STRING) = "Compaq Analyze"
    report_handle (STRING) = "mdDeCOR::gen5766"      [1]
    component_id (UINT64) = 18374966855287635968
    component_type (UINT8) = 9
    component_subtype (UINT8) = 35
    module_id (UINT64) = 11529776898687173375
    module_type (UINT8) = 21
    module_subtype (UINT8) = 35
    _hwid (UINT64) = 2
    _hwcomponent (STRING) = "CPU0"
    previous_probability (INT32) = 0
    previous_state (STRING) = "unknown"
```

[1] This `report_handle` value can be matched up with other events when there is more than one indictment per incident.

Example 3–2: Indictment Event (medium probability)

```
.
.
Formatted Message:
    Component State Change: Component "" has been indicted with a ‘medium‘
    probability of fault (HWID=0, FRUID=11962686508084822783)

Event Data Items:
    Event Name        : sys.unix.hw.state_change.indicted.medium._hwid.0
    Cluster Event     : True
    Priority          : 400
    PID               : 546043
    PPID              : 524289
    Event Id          : 958
    Member Id         : 1
    Timestamp         : 08-May-2001 15:47:08
    Host IP address   : 16.69.242.74
    Host Name         : wild-one
    Cluster Name      : wild-bunch
    Format            : Component State Change: Component
"$_hwcomponent" has
                        been indicted with a ‘medium‘ probability of
fault
                        (HWID=$_hwid, FRUID=$module_id)
    Reference         : cat:evmexp.cat:800

Variable Items:
    current_state (STRING) = "indicted"
    urgency (INT32) = 8
    probability (INT32) = 50
    total_indictments (INT32) = 2
    description (STRING) =
            "Excessive Correctable Memory Istream/Dstream Errors
reported by
             CPU0, CPU Slot0 in SoftQBB0 (HardQBB0)"
    initiator (STRING) = "Compaq Analyze"
    report_handle (STRING) = "mdDeCOR::gen5766"     1
    component_id (UINT64) = 18374966859431673855
    component_type (UINT8) = 7    2
    component_subtype (UINT8) = 38
    module_id (UINT64) = 11962686508084822783
    module_type (UINT8) = 21
    module_subtype (UINT8) = 38
    _hwcomponent (STRING) = ""
    _hwid (UINT64) = 0
    category (STRING) = ""
    previous_probability (INT32) = 0
    previous_state (STRING) = "unknown"
```

1 This `report_handle` value can be matched up with other events when there is more than one indictment per incident.

2 The value 7 for `component_type` indicates that the QBB backplane has been called out as a possible failure suspect.

Applications can be programmed to subscribe to indictment events. Example code showing how to write code that subscribes to EVM events is supplied in `/usr/examples/evm/evm_ex_olar_mon.c`.

3.1.3 Indictment Status

Enter the following command to look for components that have a non-good/non-normal status, including indicted components:

```
# hwmgr -status component -ngood

                   STATUS   ACCESS                          INDICT
 HWID:  HOSTNAME   SUMMARY  STATE              STATE        LEVEL   NAME
-------------------------------------------------------------------------------
  113:  provolone  critical online             available    high    CPU10
  194:  provolone  critical online             available    high    CPU7
```

If there is no output, then all components are in a normal state. If there is a value in the `INDICT LEVEL` column, the component has been indicted with that indictment probability. See Section 3.1.4 for more information.

Enter the following command, using the *HWID* value, to view more detailed indictment status including the urgency of the indictment:

```
# hwmgr -get attr -id HWID | grep indict
  indicted = 1
  indicted_probability = 100
  indicted_urgency = 5
```

The `indicted_urgency` attribute is a value from 1-10, the lower the value, the less urgent the removal of the component. A value of 10 indicates that you should remove the component as soon as possible.

To view the indictment information using SysMan Station, see Section 4.7.

3.1.4 Indictment Probability and Urgency

Every indictment notification has an associated probability value and a corresponding indict level. The probability value indicates the likelihood that the component being indicted is at fault. The lower the probability value, the less likely that the component is at fault.

Compaq Analyze may indict more than one component if it cannot pinpoint which component is the source of a given error. The probability value is not a true percentage likelihood of probability of future failure, but simply a method of pointing to the relative likelihood of a potential for failure.

A summary of the indictment probability values and the corresponding indict levels is shown in Table 3–1.

Table 3–1: Indictment Probability

Probability	Indict Level	Description
100	High	The most likely source of the error
50	Medium	The second most likely source of the error
1	Low	The least likely source of the error.

If this situation arises, the indictment events can be linked together by examining the report_handle variable within the indictment events. Multiple indictment events for the same error will contain the same `report_handle` value. Example 3–3 shows an example event.

The urgency is expressed as an integer value between 1 and 10. The lower the value, the less urgent the removal of the component. An urgency of 10 means the indicted component should be replaced as soon as possible. An urgency of 1 means the indicted component more than likely will fail at some future time but operator intervention may not be required immediately. The `indicted_urgency` attribute can be checked by viewing the event or with the `hwmgr` command as discussed in Section 3.1.3.

Example 3–3: Memory Indictment Event

```
Formatted Message:
    Component State Change: Physical address 268435456 has been indicted

Event Data Items:
    Event Name        :
sys.unix.hw.state_change.indicted.memory_page._physical
                        _address.268435456._hwid.0
    Cluster Event     : True
    Priority          : 200
    PID               : 530236
    PPID              : 530233
    Event Id          : 947
    Member Id         : 1
    Timestamp         : 05-Mar-2001 15:34:24
    Host IP address   : 16.69.242.74
    Cluster IP address: 16.69.241.125
    Host Name         : provolone
    Cluster Name      : deli
    Format            : Component State Change: Physical address
                        $_physical_address has been indicted
    Reference         : cat:evmexp.cat:800

Variable Items:
    current_state (STRING) = "indicted"
    urgency (INT32) = 8
    probability (INT32) = 100
    total_indictments (INT32) = 1
    description (STRING) =
            "Excessive Read Correctable Errors reported by Memory
Module0 in
             SoftQBB0 (HardQBB0)"
    initiator (STRING) = "Compaq Analyze"
    report_handle (STRING) = "mdDeCOR::gen7853"    1
    _physical_address (UINT64) = 268435456
    _hwid (UINT64) = 0
    previous_state (STRING) = "unknown"

=========================================================================
```

1 This report_handle value can be matched up with other events when there is more than one indictment per incident.

3.1.5 Clearing a Component Indictment

There are two situations when it is necessary to clear the indicted state of a component.

- The failed component has been replaced with a working component.
- Multiple components have been indicted and you need to clear the indicted state for components known to be functioning correctly.

When a component has been serviced due to an indictment, you must clear the indicted state after you verify that the repaired component is operating

properly. The indictment state is associated with the CPU slot, not the specific CPU module.

When a component has been serviced as the result of a previous component indictment, it is necessary to clear the indicted state when it has been verified that the repaired or replaced component is operating properly. Note that in the case of CPU indictments, indictment variables are associated with the CPU slot, not the specific CPU module. Therefore, when the newly replaced CPU module is inserted in a slot previously associated with an indicted CPU, it will still appear as indicted. After the newly replaced CPU module has its power on, and is verified as operating properly, you can clear the indicted state associated with the CPU slot.

Enter the following command to clear the indictment value:

```
# hwmgr -unindict [component] -id hardware-component-ID [-member cluster-member-name]
```

For example, do the following:

```
# hwmgr -unindict -id 58
```

3.2 Automatic Deallocation of Components

The Automatic Deallocation facility of the operating system subscribes to the EVM events for component indictment and can take action immediately on receipt of the notification. CPUs and memory pages that have been indicted can be taken off line automatically if wanted. Automatic deallocation behavior is defined by the variables and attributes defined in the `olar.config` and `olar.config.common` files located in the `/etc` directory. The `olar.config` file is used to define system specific policies and the `olar.config.common` is used to define cluster-wide policies. The `olar.config` file is a context-dependent symbolic link (CDSL) that is specific to the particular cluster member. Any settings in a system's `olar.config` override cluster-wide policies in the `olar.config.common` file for that system only. The values of the variables defined in this file are case insensitive.

When the Automatic Deallocation facility is invoked as a result of a component's indictment, it will post the results of its execution, including specific policy variable evaluation, as one or more EVM events. This provides an audit trail for this automated process and allows user applications to listen for (or subscribe to) these events if wanted. See `EVM(5)` for general information on the Tru64 UNIX Event Management facility. All Automatic Deallocation Facility EVM events have a prefix of `sys.unix.sysman.auto_deallocate`.

3.2.1 Automatic Deallocation Policy for CPUs

The following are policies that you can set for automatically handling CPU indictments:

- Whether or not to deallocate a CPU when it is indicted
- Time window in which to allow deallocation
- Indictment probability to allow automatic deallocation
- User-supplied script to run before deallocation
- Whether to deallocate when processes are bound

Automatic deallocation should be disabled whenever the `pfm` or `pcount` device drivers are configured into the kernel, or vice versa. For more information on these drivers, see Section 4.4.

CPU Policy Variables

The following sections describe the policy variables defined for automatic deallocation of a CPU.

Whether or Not to Deallocate CPU When Indicted

The default action is for CPUs not to be deallocated upon component indictment.

You can specify whether or not to automatically deallocate a CPU when it is indicted with the `cpu_deallocate_allow` variable. If this variable is left NULL or specified as `FALSE`, there is no automatic deallocation attempt of hardware components that belong to category CPU when a CPU is indicted. All other `cpu_deallocate*` policy variables will not be considered if this attribute is not set to `TRUE`. Allowed values are `TRUE`, `FALSE`.

Time to Perform Deallocation

You may decide to allow deallocation of a component only within a specified time window. Settings available to limit the times at which deallocation is allowed are described.

There are two variables that can be set in order to specify a time window in which automatic deallocation of indicted CPUs can take place: `cpu_deallocate_start_time` and `cpu_deallocate_end_time`. The variable `cpu_deallocate_start_time` denotes the time (in 24 hour format) beginning with and after which automatic deallocation is allowed. If no start value is specified, a value of `00:00` is assumed. Allowed values are `00:00 - 23:59`.

The variable `cpu_deallocate_end_time` denotes the time (in 24 hour format) up to and including when automatic deallocation is allowed. This

attribute is used in conjuction with `cpu_deallocate_start_time`. The start and end times are allowed to cross a day boundary. If no end value is specified, a value of `23:59` is assumed. Allowed values are `00:00 - 23:59`.

Indictment Probability for Automatic Deallocation

You can specify a single indicted probability or list of indicted probabilities for which automatic deallocation should occur for an indicted CPU using the variable `cpu_deallocate_probability`. Probabilities can be any combination of the three discrete values `low`, `medium` and `high`. Probabilities must be enclosed in braces and multiple probabilities must be delimited by a comma (,). If no value is specified for this attribute, automatic deallocation will occur only for components indicted with a high probability. Allowed values are `low`, `medium` and `high`.

Script to Run Before Deallocation

You can specify a script that can be executed before a deallocation is attempted using the `cpu_deallocate_user_supplied_script` variable. This variable contains the full path to a user-supplied script. If present, this script must be executable, be owned by root, and provide a zero return status to indicate successful execution. The script is passed two parameters that can be used in the script; the CPU name and the hardware ID (HWID) value. A non-zero return value of the script prevents the automatic deallocation from proceeding.

Whether to Deallocate Even When Processes Are Bound

This variable `cpu_deallocate_if_bound_processes` defines whether automatic deallocation of a CPU should occur if processes have been bound to run specifically on the indicted CPU or processes have been bound to the Resource Affinity Domain (RAD) that the CPU belongs to. If the value of this policy variable is TRUE, the CPU is removed automatically from the operating system (put off line) even under the following situations:

1. Processes are bound to run specifically on the CPU. Those bound processes will suspend until the CPU is brought back to the online state.
2. Processes are bound to the Resource Affinity Domain (RAD) that the CPU belongs to and this CPU is the last active CPU in the RAD. Those processes that are bound to the RAD will suspend execution until any of the CPUs that belong to the RAD are brought back to the online state.

Conversely, if this policy variable is not set to TRUE, an indicted CPU will not be deallocated if processes are bound to the CPU or if processes are bound to the RAD, which contains the indicted CPU and the CPU is the last active CPU in that RAD.

See rad_bind_pid(3) and runon(1) for information on binding processes to CPUs or to RADs.

CPU Policy Examples

The following are examples of how the variables in olar.config or olar.config.common may be set to achieve the wanted results.

Deallocate indicted CPUs immediately whenever they occur, including if processes are bound to a CPU or RAD:

```
cpu_deallocate_allow=TRUE
cpu_deallocate_start_time=00:00
cpu_deallocate_end_time=23:59
cpu_deallocate_probability=high
cpu_deallocate_user_supplied_script=
cpu_deallocate_if_bound_processes=TRUE
```

Deallocate indicted CPUs only after 7:00 p.m. and before 5:00 a.m. if indictment probability is high or medium. Do not deallocate if processes are bound to a CPU or RAD:

```
cpu_deallocate_allow=TRUE
cpu_deallocate_start_time=19:00
cpu_deallocate_end_time=04:59
cpu_deallocate_probability={high,medium}
cpu_deallocate_user_supplied_script=
cpu_deallocate_if_bound_processes=FALSE
```

Deallocate indicted CPUs immediately if the user-defined script /var/checkcpu.sh returns successfully. Do not deallocate if processes are bound to a CPU or RAD:

```
cpu_deallocate_allow=TRUE
cpu_deallocate_start_time=00:00
cpu_deallocate_end_time=23:59
cpu_deallocate_probability=high
cpu_deallocate_user_supplied_script=/var/checkcpu.sh
cpu_deallocate_if_bound_processes=FALSE
```

3.2.2 Automatic Deallocation Policy for Memory

Memory locations that have been noted by Compaq Analyze as having too many errors can be indicted. The memory page (as defined by the Page Frame Number) that contains an indicted memory location may be deallocated for use by the operating system.

Compaq Analyze can identify a physical memory location that is experiencing a high incidence of correctable single-bit errors, such that Compaq Analyze believes the error rates to be outside of normal operation. In this case, the physical location will be indicted, which may result in the memory page, containing that location, to be deallocated automatically.

If the memory page is not currently in use, then it will be mapped out (marked as bad) the next time an attempt is made to allocate the page. If

the memory page is currently in use, it will be mapped out the next time the page is deallocated.

The default setting is for a memory page to be mapped out upon indictment. Actual deallocation will occur only when the memory page is freed or subsequent access is attempted.

The following are policies that you can set for handling memory indictments:

- Whether to attempt deallocation
- Time to perform deallocation
- Probability to deallocate
- User-defined script to run on deallocation attempt

Memory Policy Variables

The following sections describe the policy variables defined for automatic deallocation of a memory page (PFN).

Whether or Not to Attempt Deallocation

You can specify whether or not automatic deallocation is allowed when a memory page is indicted with the `pfn_deallocate_allow` variable. If this attribute is left NULL or specified as `FALSE`, then there will be no automatic deallocation of memory pages when a memory page is indicted. All other `pfn_deallocate*` policy variables will not be considered if this attribute does not have the value `TRUE`. Allowed values are `TRUE`, `FALSE`.

Time Window to Perform Deallocation

You can specify the time (in 24 hour format) beginning with and after which automatic deallocation is allowed with the `pfn_deallocate_start_time` variable. This attribute is used in conjuction with `pfn_deallocate_end_time` to denote a time window in which automatic deallocation of indicted memory pages can take place. If no start value is specified, a value of `00:00` is assumed. Allowed values are `00:00 - 23:59`.

The `pfn_deallocate_end_time` variable denotes the time (in 24 hour format) up to and including which automatic deallocation is allowed. The start and end times are allowed to cross a day boundary. If no end value is specified, a value of `23:59` is assumed. Allowed values are `00:00 - 23:59`.

Probability to Deallocate

You can specify the probability values for which automatic deallocation should occur for an indicted memory page with the `pfn_deallocate_probability` variable.

This value refers to probabilities and could be any combination of the three discrete values low, medium, and high. Probabilities must be enclosed in braces and multiple probabilities must be delimited by a comma (,). If no value is specified for this attribute, automatic deallocation will occur only for memory pages indicted with a high probability. Allowed values are `low`, `medium` and `high`.

User-defined Script to Run on Deallocation Attempt

You can specify a script that will run before a deallocation attempt using the `pfn_deallocate_user_supplied_script` variable.

This variable defines a full path to a user-supplied script that will execute prior to automatic deallocation of an indicted Page Frame Number (PFN). If present, this script must be executable, be owned by root, and provide a zero return status to indicate successful execution. The script is started with a parameter that can be used in the script, the decimal value of the PFN. A non-zero return value of the script will prevent the automatic deallocation from proceeding.

Memory Deallocation Examples

The following are examples of how the variables in `olar.config` or `olar.config.common` may be set to achieve the wanted results for memory deallocation.

Deallocate indicted memory pages immediately whenever indictment events occur:

```
pfn_deallocate_allow=TRUE
pfn_deallocate_start_time=00:00
pfn_deallocate_end_time=23:59
pfn_deallocate_probability=high
pfn_deallocate_user_supplied_script=
```

Deallocate indicted memory pages only after 7:00 p.m. and before 5:00 a.m.:

```
pfn_deallocate_allow=TRUE
pfn_deallocate_start_time=19:00
pfn_deallocate_end_time=04:59
pfn_deallocate_probability=high
pfn_deallocate_user_supplied_script=
```

Deallocate indicted memory pages immediately if the user-defined script `/var/checkmem.sh` returns successfully:

```
pfn_deallocate_allow=TRUE
pfn_deallocate_start_time=00:00
pfn_deallocate_end_time=23:59
pfn_deallocate_probability=high
pfn_deallocate_user_supplied_script=/var/checkmem.sh
```

4

Online Addition and Removal

This chapter discusses the features of the operating system that support this process including the following topics:

- Reasons you may want to use OLAR (Section 4.2)
- Getting component state information (Section 4.3)
- Cautions and restrictions for OLAR operations (Section 4.4)
- How to add components (Section 4.5)
- How to remove components (Section 4.6)
- Using SysMan Station to monitor and manage OLAR (Section 4.7)

4.1 Overview

This chapter describes how to add or replace components in a system while keeping an operating system instance and associated applications running.

Online Addition and Removal (OLAR) management is provided by the `hwmgr` command and the SysMan suite of System Management applications. Complete management is available from the operating system, eliminating the need to use lower level hardware monitor interfaces such as the System Reference Monitor (SRM) or System Control Monitor (SCM).

4.2 Reasons for Component OLAR

Online Addition and Removal (OLAR) management allows for the addition or removal of hardware while the operating system and applications continue to run. This provides the benefit of increased system up time and availability during both scheduled and unscheduled maintenance. OLAR is supported for CPUs on some symmetrical multiprocessing (SMP) platforms. Currently, the platforms which support CPU OLAR are the AlphaServer GS160, and GS320 series systems. Other SMP systems do not support physically adding or removing CPUs while the system is running, but do support placing a CPU in an offline state if it is not functioning properly.

The need for component OLAR may arise for one of the following reasons:

Computation Capacity Expansion	A system requires additional computational resource capacity. For example, a GS320 may have increased

	processing requirements. If the system has available CPU slots, the CPU capacity can be expanded by adding additional CPU modules to the system to improve system performance.
Maintenance Upgrade	A system manager wants to upgrade specific system components to the latest model or revision. As an example, a GS160 with earlier model CPU modules can be upgraded to later model CPUs with higher clock rates, while the operating system continues to run. In this example for GS series systems, all CPUs in a Quad Building Block (QBB) must be running the same model and speed CPU.
Failed Component Replacement	A system component is indicating a high incidence of correctable errors and the system manager wants to perform a proactive replacement of the failing component before it results in a hard failure.

4.3 Getting State Information

You can get information about components and their states using the `hwmgr` command, SysMan Station, or by viewing particular events. The following sections discuss information pertaining to component states during an Online Addition and Removal operation.

4.3.1 Component States and Status

There are three important attributes, which describe how a component is currently functioning:

- Access State
- State
- Status
- Indicted

These attributes can be displayed by using the `hwmgr -status` command or by viewing the properties of a component with the SysMan Station.

For information on how to view the properties using SysMan Station, see Section 4.7.

Access State Attribute

The access state attribute of a component is an indication of the accessibility of a component to the operating system, as determined by a system administrator. The access state of a component is either on line or off line.

An online component is used actively by the operating system. An offline component is not used by the operating system. For example, offline CPUs will not have processes scheduled for execution by them.

State Attribute

The state attribute, in general, is an indication of the component's operational capabilities, as indicted by the controlling software for a given component. The following are the possible states of the components:

Available	The component is fully functional and ready for use although it might not be currently on line.
Unavailable	The component is unavailable for use.
Off	The component is turned off.
Unknown	The controlling software is unable to determine the status of the component. Use other `hwmgr` command options and diagnostic or service tools to determine its status.

Status Attribute

The status attribute is a summary of the access state, state and indicted state attributes, to provide a quick indication of the component status. The component status is one of the following:

Normal/Good	The component is behaving normally.
Inactive	The status of the component is inactive because it is a component that is managed using the Compaq Capacity on Demand (CCoD) feature (typically a CPU). The component is physically present but off line and therefore available for spare capacity.

Warning	This status warns you that a component is not in a normal state but may return to a normal state after a system reboot. For example, when you take a CPU off line using the `-offline nosave` option, its status changes to warning state. It is considered a warning status because this CPU automatically will become on line and available after system reboot or initialization.
Critical	This status warns you that a component is not in a normal state and will not return automatically to a normal state. You must intervene to bring the component back to a normal state (on line and available). For example, when you take a CPU off line, its offline state persists across a reboot and its status changes to critical. You only can bring the CPU back on line by manual intervention. Other examples of components that will cause a critical status are components that are indicted (through the Component Indictment facility), and components with power off.

Indicted Attribute

The indicted attribute is an indication of whether a component has been indicted by a fault analysis utility. If a component has been indicted, the additional attributes `indicted_probability` and `indicted_urgency` are also set. For more information on these attributes, see Section 3.1.4.

4.3.2 OLAR Events

OLAR operations will cause a change in a component's state. All changes in a component state will result in the generation of an EVM event. EVM events that track changes in the state of a component begin with `sys.unix.hw.state_change`. Events that result from OLAR operations are hardware state change events. For a description of each type of state change event that can occur, enter the following command:

```
# evmwatch -i -f  '[name sys.unix.hw.state_change]'  | evmshow -t "@name" -x | more
```

See `EVM`(5) or the *System Administration* manual for more information.

Applications can be programmed to subscribe to OLAR events. Examples showing how to write code that subscribes to OLAR specific EVM events is supplied in `/usr/examples/evm/evm_ex_olar_mon.c`. You must have the `OSFEXAMPLES`*nnn* subset installed to get the /usr/examples directory.

4.3.3 Locating a CPU

If you are removing a CPU in an AlphaServer GS160/GS320, you may need to locate the Quad Building Block (QBB) number for each installed CPU. If you are using SysMan Station, the Hardware View window shows the hierarchy of the hardware graphically. If you are using a command line interface, enter the following command to get information on the hardware hierarchy. `hwmgr -view hierarchy`:

```
# hwmgr -view hierarchy
```

```
 HWID:   hardware hierarchy     (!)warning (X)critical (-)inactive (see -status)
 ------------------------------------------------------------------------------
    1:   platform Compaq AlphaServer GS160 6/731
    9:     bus wfqbb0
   10:       connection wfqbb0slot0
   11:         bus wfiop0
   12:           connection wfiop0slot0
   13:             bus pci0
   14:               connection pci0slot1

     .
     .

   57:       cpu qbb-0 CPU0 [1]
   58:       cpu qbb-0 CPU2
```

[1] This line shows the hardware ID (`57`), the component type (`cpu`), the hard Quad Building Block (QBB) number where the CPU is located (`qbb-0`), and the CPU name (`CPU0`). Note that the hard QBB number does not change in a partitioned system.

To quickly identify which QBB a CPU is associated with, using the CPU hardware ID, enter the following command:

```
# hwmgr -view hier -id HWID
```

```
 HWID:   hardware hierarchy
 ------------------------------------------------------------------------------
    58:   cpu qbb-0 CPU0
```

4.4 Cautions Before Performing CPU OLAR Operations

The following cautions must be considered before adding or removing CPUs:

- Applications potentially may suspend if they bind threads to a particular CPU and that CPU is taken off line. If you are running such an application on a cluster member that is being serviced, you may want to temporarily relocate that application to another member. Also, any CPU with processes bound to a RAD of which this is the last running CPU should not be put off line unless it is acceptable to suspend processes that are bound to that RAD.

- You must have root privileges or have the appropriate DOP privileges to use the Manage CPUs application. The DOP privilege for the Manage CPU application for OLAR management is `HardwareManagement`. To use the `hwmgr` command, you must be logged in as the root user. Only one administrator (with root privileges) at a time can initiate OLAR operations; other administrators will be prevented from initiating OLAR operations momentarily.
- If you are using program profiling utilities such as `dcpi`, `kprofile`, or `uprofile`, that are aware of the system's CPU configuration, unpredictable results may occur when performing OLAR operations. It is therefore recommended that these profiling utilities be disabled prior to performing an OLAR operation. Ensure that all the processes including any associated daemons that are related to these utilities have been stopped before performing OLAR operations on system CPUs. The device drivers used by these commands usually are configured into the kernel dynamically, so the commands can be disabled before each OLAR operation with the following commands:
  - `sysconfig -u pfm`
  - `sysconfig -u pcount`
- You reenable the appropriate device driver by using the following commands:
  - `sysconfig -c pfm`
  - `sysconfig -c pcount`

    The automatic deallocation of CPUs, enabled through the Automatic Deallocation Facility, should be disabled whenever the `pfm` or `pcount` device drivers are configured into the kernel, or vice versa.

    See `sysconfig`(1) and `dcpi`(1) for more information. Documentation for the Compaq Continuous Profiling Infrastructure (DCPI) applications can be found at **http://www.tru64unix.compaq.com/dcpi/documentation.htm**.
- You cannot put a CPU off line if it is the only CPU in the default processor set (0). To verify which processor set a processor is in, look at the `psed_id` field using the `pset_info`(8) command. See `pset_info`(8) for more information.
- If you have CPUs installed in Quad Building Blocks that do not have memory installed, none of these CPUs can become the primary processor if you attempt to put the existing primary processor off line. The primary processor therefore cannot be put off line if the only active CPUs available are located in Quad-Building Blocks with no memory even if they are in the default processor set (0).

- If a process has been specifically bound to execute on a CPU, you must decide how to handle the process. See `runon`(1), `bind_to_cpu`(3), and `bind_to_cpu_id`(3) for more information. If an OLAR operation is attempted on that CPU, you will be notified by the OLAR utilities that processes have been bound to the CPU prior to any operation being performed. You may choose to continue or cancel the OLAR operation. By choosing to continue, processes bound to a CPU will suspend their execution until such time that the process is not bound, or the CPU is placed back on line. Choosing to put off line a CPU that has processes bound may cause detrimental consequences to the application, depending upon the characteristics of the application.
- If a process has been specifically bound to execute on a Resource Affinity Domain (RAD) you must decide how to handle the process. See `runon`(1), and `rad_bind_pid`(3) for more information. If an OLAR operation is attempted on the last running CPU in the RAD, you will be notified by the OLAR utilities that processes have been bound to the RAD and that the last CPU in the RAD has been requested to be placed off line. By choosing to continue, processes bound to the RAD will suspend their execution until such time that the process is unbound, or at least one CPU in the RAD is placed on line. Note that choosing to put off line the last CPU in a RAD with processes bound may cause detrimental consequences to the application, depending upon the characteristics of the application.

4.5 Component Removal Procedure

The process of removing a component consists of the following steps:

1. Use one of the appropriate management applications (`the hwmgr` command or SysMan) to prepare for the removal of a component, such as verifying the status of the CPU and ensuring that no user processes are bound to it currently. Any processes that are not specifically bound to a CPU are migrated automatically to other running CPUs when the CPU is put off line. Also, any CPU with processes bound to a RAD of which this is the last running CPU should not be put off line unless it is acceptable to suspend processes that are bound to that RAD.
2. Use one of the appropriate management applications to take the component off line and remove power. See Section 4.5.1 for more information. When power is removed, the LED on the CPU module will illuminate yellow, indicating that the CPU module power is off and it is safe to remove.
3. Remove the component physically. The operating system automatically recognizes that the CPU module physically has been removed. There

is no need to perform a scan operation to update the hardware configuration.

Before a CPU can be removed physically from the system, it must be placed off line and the power turned off, using any of the supported management applications described in the following sections. Processes queued for execution on a CPU that is to be placed off line simply are migrated to run-queues of other running (online) processors.

If another system administrator is actively managing the systems' processors, you will get a warning message telling you to perform the operation at another time.

4.5.1 Taking CPUs Off Line, Removing Power, or Both

You may want to take a CPU off line if it is suspected of potentially failing. Compaq Analyze proactively may indicate that a CPU is suspected of a potential failure by notifying the Component Indictment facility, which will create EVM events about the indictment. For information on component indictment, see Section 3.1.

A CPU that is placed off line will be persistently off line across reboots, by default. You optionally may set a CPU to be put on line at the next reboot.

On AlphaServer GS80, GS160, or GS320 systems or ES45 systems, any CPU can be placed off line as long as it is not the last CPU in the primary processor set. On other older SMP systems, any CPU except the primary CPU can be placed off line. If your system supports OLAR of CPUs, you also can turn the power off to allow for removal of the CPU.

If OLAR is not supported on your SMP system, the CPU may be placed off line, but may not have power turned off by the operating system. This will stop scheduling of processes on this processor and potentially avoid a kernel panic if the processor is experiencing uncorrectable errors.

If your SMP system does not support OLAR, the Manage CPUs application will not offer you the opportunity to remove power from the CPU, and the `hwmgr` command will return an appropriate message if you attempt to remove power from the CPU. Attempts to use the `hwmgr` command to remove power from a CPU in a system that does not support OLAR will not succeed and an error message will be displayed.

4.5.2 Taking CPUs Off Line, Removing Power, or Both Using SysMan Menu

For instructions on how to start the SysMan Menu, see the *System Administration* manual.

To take a CPU off line using the SysMan Menu, do the following:

1. Select Hardware.
2. Select Manage CPUs in the SysMan Menu application. Only one system administrator can put a CPU off line at any given time.
3. Select the CPU or the CPUs that you want to put off line. For instructions on selecting multiple CPUs, see the online help.
4. Select Modify ....
5. If the system does not support OLAR, select Off line. If the system supports OLAR, select one of the following:
   - Off line (powered off) - place the CPU off line and remove power
   - Off line (powered on) - place the CPU off line and keep the CPU supplied with power

   By default, the CPU remains in the state you chose for subsequent system reboots. To have the CPU automatically go on line at the next system reboot, do the following:

   a. Select Offline Options... in the Manage CPUs Modify dialog box.
   b. Select the Bring selected CPUs on line at system reboot checkbox.
   c. Select OK in the Offline Options dialog box.
6. Select OK to complete the offline operation.

See the Manage CPUs Online Help for additional information on performing offline operations.

4.5.3 Taking CPUs Off Line, Removing Power, or Both Using SysMan Station

For instructions on how to start the SysMan Station, see the *System Administration* manual.

To take a CPU off line using the SysMan Station, do the following:

1. Select the Hardware View window.
2. Select a CPU icon and press MB1.
3. Select Manage CPUs from the Tools menu. Follow the steps in Section 4.5.2.

See the Manage CPUs Online Help for additional information on performing offline operations.

4.5.4 Taking a CPU Off Line Using the hwmgr Command

To take a CPU Off line using the `hwmgr` command, do the following:

1. Verify the status of the component. The access state of a CPU must be on line. Note the HWID number and name of the CPUs for use in later steps. Enter the following command:

   # hwmgr -status component

```
                  STATUS   ACCESS                          INDICT
 HWID:  HOSTNAME  SUMMARY  STATE              STATE        LEVEL   NAME
--------------------------------------------------------------------------------
   .
   .
   .
   57:  wild-one           online             available            CPU0
   58:  wild-one  critical offline            available            CPU2
   59:  wild-one           online             available            CPU4
   60:  wild-one           online             available            CPU6
```

2. Enter either of the following commands to put the component off line:

   # /usr/sbin/hwmgr -offline -name *cpu-name*

   # /usr/sbin/hwmgr -offline -id *HWID*

   If the component is unable to be put off line due to processes bound to the CPU, or if the processor is the last processor in the primary processor set, the command will notify you and suggest using the `-force` option after you assess the impact of putting the CPU off line.

   Additional options of [`-nosave`] or [`-force`] can be used. The CPU [`-nosave`] option specifies that on the next system reboot the CPU will be brought back on line. The [`-force`] option forces a CPU off line if processes were identified as being bound to that CPU.

   If you now want to remove power from the component, see Section 4.5.5.

4.5.5 Removing Power from CPUs Using the hwmgr Command

To remove power to an offline CPU using the `hwmgr` command, do the following:

1. Verify the status of the component. The access state of a CPU must be off line in order to remove power from it. If you need to take a CPU off line, see Section 4.5.4. Enter the following command to view the component status.

   # hwmgr -status component

```
                  STATUS   ACCESS                          INDICT
 HWID:  HOSTNAME  SUMMARY  STATE              STATE        LEVEL   NAME
--------------------------------------------------------------------------------
.
.
.
   57:  wild-one           online             available            CPU0
```

```
   58:  wild-one   critical offline           available           CPU2
   59:  wild-one            online            available           CPU4
   60:  wild-one            online            available           CPU6
```

To limit the view to only components that have a status summary value other than good, enter the command:

```
# hwmgr -status component -ngood
```

```
                  STATUS   ACCESS                          INDICT
 HWID:  HOSTNAME  SUMMARY  STATE              STATE        LEVEL   NAME
------------------------------------------------------------------------------
.
.
.
   58:  wild-one   critical offline           available           CPU2
```

2. Use the CPU's HWID value to verify if the CPU is able to have its power turned off. Enter the following command:

```
# hwmgr -get attribute -a capabilities -id 58
```

```
58:
  capabilities = 1
```

If the capabilities value is 1, the CPU is capable of having its power turned off. If the capabilities value is 0, the CPU cannot have its power turned off.

3. Enter either of the following commands to remove power to the component:

```
# /usr/sbin/hwmgr -power off -name cpu-name
```

```
# /usr/sbin/hwmgr -power off -id HWID
```

4.6 Component Addition Procedure

The process of inserting a CPU module component consists of the following steps:

1. Add the component physically. Select an available CPU slot in one of the configured Quad Building Blocks (QBB). If there are available slots in several QBBs, it is typically best to equally distribute the number of CPUs among the configured QBBs.

   Insert the CPU module into the CPU slot. Ensure that you align the color-coded decal on the CPU module with the color-coded decal on the CPU slot. The LED on the CPU module will illuminate yellow, indicating that the CPU module's power is off. Note that the CPU will be recognized automatically by the operating system, even though it does not yet have power applied. There is no need to perform a scan operation for the operating system to identify the CPU module.

Warning

You should not add a component without referring to the component documentation, which contains important safety information and information on preventing static discharges that can destroy the component.

2. Use one of the appropriate management applications (the `hwmgr` command or SysMan) to apply power to the component and place it on line. See Section 4.6.1 for more information. When power is applied to the CPU, it will undergo a short self-test (7-10 seconds), after which the LED will illuminate green, indicating the CPU module power is on and has passed its self-test. When the CPU is placed on line, the operating system will begin automatically to schedule and execute tasks on this CPU.
3. Use one of the appropriate management applications to verify that the component is functioning properly. If the CPU is a replacement for one that has been indicted, be sure to clear the indictment after the component has been verified as functioning properly, as discussed in Section 3.1.5.

Newly inserted CPUs are recognized automatically by the operating system, even before their power is on. They cannot start scheduling and executing processes until the CPU has power on and is placed on line.

If another system administrator is actively managing the systems' processors, you will get an error message telling you to perform the operation at another time.

Warning

You must follow all safety procedures as documented in the hardware documentation accompanying the component. You also should consult the component replacement procedures in the service manual for your system. Failure to follow safety procedures could result in personal injury or could damage the component.

4.6.1 Putting CPUs On Line, Applying Power, or Both

You may place a CPU on line if its access state is off line and its state is available. This typically is done when a CPU is replaced or newly installed.

4.6.2 Putting CPUs On Line, Applying Power, or Both Using SysMan Menu

For instructions on how to start the SysMan Menu, see the *System Administration* manual.

To put a CPU on line using the SysMan Menu, do the following:

1. Select Hardware.
2. Select Manage CPUs in the SysMan Menu application.
3. Select the CPU or the CPUs that you want to put on line.
4. Select Modify ....
5. If the system does not support OLAR, select On line. If the system supports OLAR, select one of the following:
   - On line - place the CPU on line and apply power if necessary
   - Off line (powered on) - apply power to the CPU but keep it off line

   By default, the CPU will remain in the state you chose for subsequent system reboots.
6. Select OK to complete the operation.

See the Manage CPUs Online Help for additional information on performing online operations.

4.6.3 Putting CPUs On Line, Applying Power, or Both Using SysMan Station

For instructions on how to start the SysMan Station, see the *System Administration* manual.

To put a CPU on line using the SysMan Station, do the following:

1. Select the Hardware View window.
2. Select a CPU icon and press MB1.
3. Select Manage CPUs from the Tools menu. Only one system administrator can place a CPU on line at any given time.

Follow the steps in Section 4.6.2.

See the Manage CPUs Online Help for additional information on performing online operations.

4.6.4 Applying Power to CPUs with the hwmgr Command

To apply power to a CPU that is off line and has power off using the `hwmgr` command, do the following:

1. Verify the status of the component. Note the ID number of the CPUs in the first column of the output. Enter the following command to view the status of the components:

   # hwmgr -status component

```
                  STATUS   ACCESS                          INDICT
 HWID:  HOSTNAME  SUMMARY  STATE              STATE        LEVEL   NAME
--------------------------------------------------------------------------------
.
.
.
   57:  wild-one           online             available            CPU0
   58:  wild-one  critical offline            available            CPU2
   59:  wild-one           online             available            CPU4
   60:  wild-one           online             available            CPU6
```

   To limit the view to only components that have a status summary value other than `good`, enter the following command:

   # hwmgr -status component -ngood

```
                  STATUS   ACCESS                          INDICT
 HWID:  HOSTNAME  SUMMARY  STATE              STATE        LEVEL   NAME
--------------------------------------------------------------------------------
.
.
.
   58:  wild-one  critical offline            available            CPU2
```

2. Enter either of the following commands to apply power:

   # /usr/sbin/hwmgr -power on -name *cpu-name*

   # /usr/sbin/hwmgr -power on -id *HWID*

The CPU must be placed on line before the operating system schedules processes to be run on the CPU. If you want to bring the CPU on line, see Section 4.6.5

4.6.5 Putting a CPU On Line Using the hwmgr Command

To put a CPU on line using the `hwmgr` command, do the following:

1. Enter the following command to verify the status of the component:

   # hwmgr -status component

```
                  STATUS   ACCESS                          INDICT
 HWID:  HOSTNAME  SUMMARY  STATE              STATE        LEVEL   NAME
--------------------------------------------------------------------------------
.
.
.
   57:  wild-one           online             available            CPU0
```

```
58:  wild-one   critical offline           available           CPU2
59:  wild-one            online            available           CPU4
60:  wild-one            online            available           CPU6
```

2. Enter either of the following commands to put the CPU on line:

   # /usr/sbin/hwmgr -online -name *cpu-name*

   # /usr/sbin/hwmgr -online -id *HWID*

4.7 Monitoring and Managing Components with SysMan Station

SysMan Station is a GUI based System Management application and is a central point from which to manage and monitor your system.

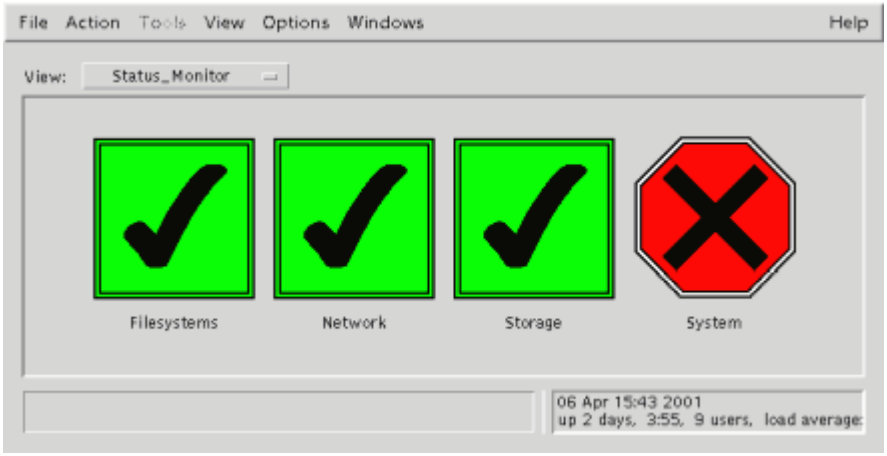
Use SysMan Station to monitor components such as CPUs. When viewing any system component, you can obtain detailed information on its properties or launch applications that enable you to perform administrative tasks on the component.

The SysMan Station cannot be used in a character cell user environment like the SysMan Menu. SysMan Station requires that your system support graphics capability.

The SysMan Station has extensive Online Help for using the application.

The Status view of the SysMan Station tells you the status of your system at a glance. The Status lights for each attention group can be green, yellow, or red. If an attention group has a green light, all is well for that part of the system. If an attention group has a yellow light, there has been an event indicating a warning for that group. A red light for one of the attention groups indicates a critical condition that requires attention. Figure 4–1 shows indicted components causing the System attention group to be red.

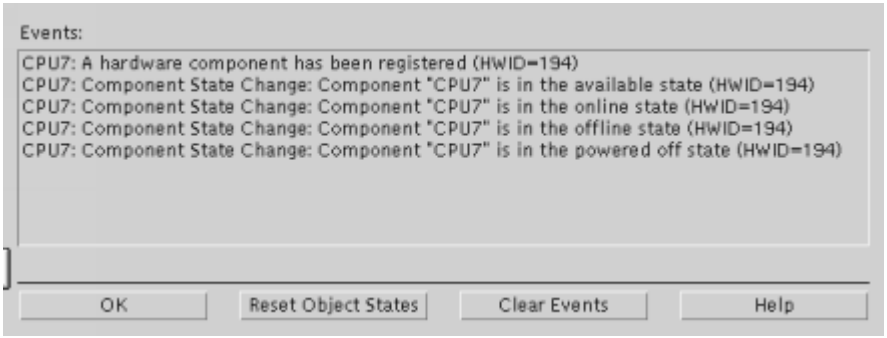
Figure 4–1: Status View



ZK-1841U-AI

If a component is indicted, you can view recent events by double clicking on the System attention group. Figure 4–2 shows an example of state change events. You can double click on any of the events to see full information on the event.

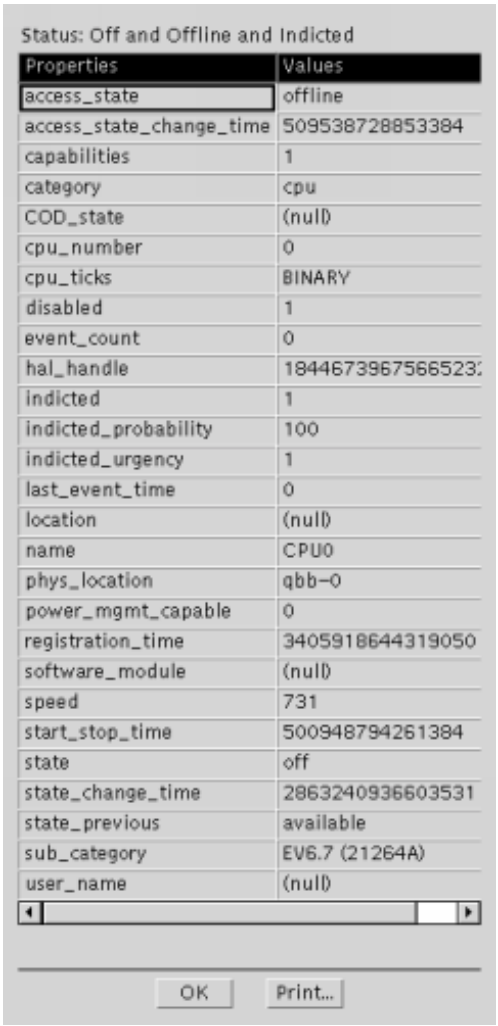
Figure 4–2: Events



ZK-1842U-AI

To view the status of a component, in the Hardware View of SysMan Station, click on the component with MB3. A window will open. Select the Properties menu item to see the current status of the component, including all values discussed in Section 4.3.1 and indictment values. Figure 4–3 shows an example of the properties of a CPU.

Figure 4–3: CPU Properties



ZK-1843U-AI

5

Service Tools

This chapter describes how to use the various service tools to help minimize system down time. Using the service tools described, service technicians can be notified of the state of your system, making it easier and quicker to recover from errors.

This chapter discusses the following topics:

- WEBES Service Tools integration with SysMan (Section 5.1)
- Compaq Analyze (CA) (Section 5.2)
- Compaq Crash Analysis Tool (CCAT) (Section 5.3)
- Revision & Configuration Management (RCM) (Section 5.4)
- The `sys_check` tool (Section 5.5)
- The `collect` tool (Section 5.6)
- Service Tools and Monitoring Applications Quick Start (Section 5.7)
- Crash Dump and Save Core Applications (Section 5.8)

5.1 WEBES Service Applications and SysMan

Web-Based Enterprise Services (WEBES) provides a web-based suite that integrates hardware analysis software, and revision management tools. The WEBES service tools are included on the operating system APCD or supplied by service personel.

The WEBES tools that are available for Tru64 UNIX are:

- Compaq Analyze (CA)
- Compaq Crash Analysis Tool (CCAT)
- Revision & Configuration Management (RCM)

Information on the WEBES service tools can be found at:

`http://www.support.compaq.com/svctools/webes/index.html`

After WEBES is installed, you can run these tools directly from the SysMan Station. In addition, RCM is available from the SysMan Menu.

See Chapter 4 for more information on the various applications and interfaces used to manage and monitor your system.

Compaq service personnel can be notified with the System Initiated Call Logging (SICL) to the service provider's customer service center through its Automatic Call Handling System. System Initiated Call Logging (SICL) is functional only if you also install DSNlink. Information on DSNlink can be found at **http://www.support.compaq.com/dsnlink**.

5.2 Compaq Analyze (CA)

Compaq Analyze can help minimize down time by providing proactive error notification of faulty hardware components. Faulty components can be automatically put off line using the Automatic Deallocation facility or manually put off line by the administrator to avoid system panics.

Compaq Analyze is a rules-based hardware fault management diagnostic application that provides error event analysis and translation. The multievent correlation analysis feature of Compaq Analyze provides analysis of events stored in the binary system event log or other specified binary log files.

By default, Compaq Analyze provides the analysis used to indict a component and notifies the operating system's component indictment facility. See Section 3.1 for more information on component indictment. Indictment support is only available in Compaq Analyze 4.0 or higher.

Compaq Analyze can be set up at installation for automatic notification of system administrators or service personel. By default, System administrators will be notified by e-mail whenever Compaq Analyze detects a faulty component. Compaq service personel also can be notified with the System Initiated Call Logging (SICL) to the service provider's customer service center through its Automatic Call Handling System. This is configureable in Compaq Analyze and described fully in the Compaq Analyze documentation.

Compaq Analyze also provides an option to run in manual mode. Manually generated analysis by Compaq Analyze does not send automatic notification by indictment, e-mail, or SICL.

When Compaq Analyze is installed, the GUI interface can be launched directly from the SysMan Station by clicking on the Host Icon and selecting Compaq Analyze from the Tools menu.

5.3 Compaq Crash Analysis Tool (CCAT)

The Compaq Crash Analysis Tool (CCAT) can help minimize down time by potentially providing a user with ways of recovering from a system crash quickly.

The CCAT tool can be configured to automatically send crash parameters or results files to the Compaq Support Center (CSC). It also can send e-mail notification to system administrators.

This tool collects data that describes system crashes and matches the data against a set of operating system specific rules to determine if the footprint of the collected crash data matches any known crash data footprints for which a solution or corrective action is known. This capability significantly reduces customer down time by shortening the time required to analyze system crashes.

The CCAT graphical user interface (GUI) is an interactive tool used to analyze crash files manually. The CCAT GUI is used only for onsite manual tasks. It does not log calls or send crash parameters or results to the CSC nor does it send e-mail notification to anyone.

When CCAT is installed, the GUI interface can be launched directly from the SysMan Station by clicking on the Host Icon and selecting CCAT from the Tools menu.

For specific installation and user details, see the *Compaq Crash Analysis Tool User* guide at **http://www.support.compaq.com/svc-tools/webes/webes_docs.html**.

5.4 Revision & Configuration Management (RCM)

The Revision and Configuration Management (RCM) tool provides revision and configuration reporting for Compaq AlphaServer systems running Tru64 UNIX. Under normal circumstances the RCM application is used by Compaq Service Engineers and Compaq Support Center specialists to collect revision and configuration data from customer systems.

The types of reports that RCM can create are as follows:

- Configuration Report - an inventory of the components on the target system, based on a single data collection.
- Change Report - shows the difference between two data collections on the same system.
- Comparison Report - shows the differences between data collections on two different systems.
- Analysis Report - can generate either of the following analysis reports:
  - Patch analysis for Tru64 UNIX Version 4.0E, 4.0F, and 4.0G systems
  - Hardware revision analysis for AlphaServer ES40 systems

After RCM is installed, it can be launched directly from the SysMan Menu by selecting the Support and Services branch and then selecting Configure the RCM Data Collector.

For specific installation and user details, see the *Revision and Configuration Management Data Collector for Compaq Tru64 UNIX* user guide at **http://www.support.compaq.com/svc-tools/webes/webes_docs.html**.

5.5 The sys_check Tool

The sys_check tool can help reduce system down time by as much as 50 percent by providing fast access to critical system data. It is recommended that you run a full check at least once a week to maintain the currency of system data. However, some options will take a long time to run and can impact system performance. You should therefore choose your options carefully and run them during off-peak hours. As a minimum, perform at least one full run (all data and warnings) as a postconfiguration task in order to identify configuration problems and establish a configuration baseline.

The sys_check tool is a System Administration application that creates an HTML file that describes the system configuration, and it can be used to diagnose serviceability problems.

The sys_check tool is a system census and configuration verification tool that also is used to aid in diagnosing system errors and problems. Use the sys_check tool to create an HTML report of your system's configuration (software and hardware).

You can run the sys_check tool from the SysMan GUI applications or from the command line interface. For further information on using the sys_check tool, see the *System Administration* manual, *System Configuration and Tuning* manual and sys_check(8).

The sys_check tool also performs an analysis of operating system parameters and attributes such as those that tune the performance of the system. The report generated by the sys_check tool provides warnings if it detects problems with any current settings. While the sys_check tool can generate hundreds of useful warnings, it is not a complete and definitive check of the health of your system. The sys_check tool should be used in conjunction with event management and system monitoring tools to provide a complete overview and control of system status. See EVM(5) for more information on event management. See the *System Administration* manual for information on monitoring your system.

Running the sys_check tool for warning information on possible configuration problems or for performance data takes less time than other options and we suggest you do so once per week.

After you perform OLAR operations, you can use the `sys_check` tool to check your system configuration. You can use the analysis information to determine if there are potential problems with the operations you just performed. The `sys_check` tool creates an HTML file that describes the system configuration, and aids you in diagnosing system errors and problems. The application checks system components such as CPUs and provides performance data for those system components. The `sys_check` tool outputs any warnings and tuning guidelines, which you can use to improve system performance.

5.6 The collect Tool

The `collect` tool is a system monitoring application that records or displays specific operating system and process data for a set of subsystems. You can configure the `collect` tool to automatically start when the system is rebooted. The `collect` tool can assist you in diagnosing performance problems and its report output is requested by your technical support service when they are assisting you in solving system problems. See `collect`(8) and the *System Administration* manual for more information.

5.7 Service Applications and Monitoring Applications Quick Start

If you are familiar with the service applications that support the operating system, you can begin using them right away. Table 5–1 summarizes the applications.

Table 5–1: Service Applications Quick Applications Start

Service Applications	Interface Used	Invoking the Application	Command Line
Compaq Analyze (CA)	SysMan Station	host icon —> Tools —> Compaq Analyze	`/usr/sbin/ca`
Compaq Crash Analysis Tool (CCAT)	SysMan Station	host icon —> Tools —> CCAT	`/usr/sbin/ccat gui`
Revision & Configuration Management (RCM)	SysMan Station and SysMan Menu	host icon —> Tools —> RCM	`unisetup`

Table 5–1: Service Applications Quick Applications Start (cont.)

Service Applications	Interface Used	Invoking the Application	Command Line
`sys_check` tool	SysMan Station and SysMan Menu	`sysman config_report` or `sysman escalation`	`sys_check -perf` or `sys_check -escalate`
`collect` tool	SysMan Station and SysMan Menu	`collect`	`/usr/sbin/collect`

5.7.1 Recommended Schedule and Use

When you use the service applications for fault diagnosis, the applications can reduce system down time and enhance system serviceability by providing fast access to critical system configuration data. Table 5–2 gives you some recommended guidelines that you can use to maintain the currency of system data. However, note that some applications will take a long time to run and can impact system performance. You therefore should choose your applications carefully and run them during off peak hours. As a minimum, perform at least one full run (all data and warnings) as a postconfiguration task in order to identify configuration problems and establish a configuration baseline.

Table 5–2 provides guidelines for balancing data needs with performance impact.

Table 5–2: Recommended Schedule and Use

Service Application	Purpose and Use of Application	Frequency of Use
Compaq Analyze (CA)	Fault analysis and Fault avoidance	Continually running
Compaq Crash Analysis Tool (CCAT)	Fault analysis	After a system crash
Revision & Configuration Management (RCM)	Generates system configuration information and analysis	As needed
`sys_check -perf -warn`	Generates system configuration information and analysis	Run weekly

Table 5–2: Recommended Schedule and Use (cont.)

Service Application	Purpose and Use of Application	Frequency of Use
`sys_check`	Generates system configuration information and analysis	Run at least once after installation and after major configuration changes
`sys_check -all`, or `-escalate`, or `-noquick`	Generates complete system configuration information and analysis	Run only when troubleshooting

5.8 Crash Dump and Save Core Commands

The `dumpsys` and `savecore` applications can help diagnose problems after a system crash.

The `savecore` command usually is invoked automatically during system startup. It determines whether a crash dump has been made, and if there is enough file system space to save it. See `savecore`(8), the *System Administration* manual or the *Kernel Debugging* manual for more information.

The `dumpsys` command copies a snapshot of memory to a dump file, without halting the system. This feature is useful for estimating crash dump size during dump configuration planning. See `dumpsys`(8), the *System Administration* manual or the *Kernel Debugging* manual for more information.

6

Memory Trolling

This chapter discusses the features of the operating system that support this process including the following topics:

- Enabling, Disabling, and Tuning Memory Trolling (Section 6.2)
- Controlling system resource use (Section 6.3)
- Memory Troller Messages (Section 6.4)
- Interactions with OLAR (Section 6.5)

6.1 Overview

The operating system handles memory errors with a just-in-time scrubbing model, where correctable errors are scrubbed when encountered by the operating system or an application. To enhance this capability, a trigger mechanism, called the memory troller proactively locates and scrubs correctable memory errors. The memory troller systematically reads each memory location. If it discovers a correctable memory error, it triggers the just-in-time scrubbing mechanism.

Because the memory troller reads all memory available to the operating system, it also might discover uncorrectable memory errors, which would lead to an unrecoverable machine check. To avoid this, the operating system recognizes that the machine check resulted from memory trolling, dismisses the error, and continues normal operation. The memory troller then causes the memory page containing the uncorrectable error to be marked as a bad page. If the bad page is free (or when it becomes free) it is then mapped out so it will not be reused.

6.2 Enabling, Disabling, and Tuning Memory Trolling

For systems supported by the memory troller, use the `vm_troll_percent` variable to enable, disable, and tune the trolling rate. This parameter is part of the kernel's `vm` subsystem. The trolling rate is expressed as a percentage of the system's total memory trolled per hour and can be changed at any time. Valid troll rate settings are as follows:

Default value: 4 percent per hour

This value is used by default if you do not specify any value for vm_troll_percent. At this default rate, each 8 kilobyte memory page is trolled once every 24 hours.

Disable value: 0 (zero)

A value of zero disables the memory troller.

Range: 1 - 100 percent

The troll rate is set to the specified percentage of memory to troll per hour. For example, a 50 percent troll rate reads half the total memory in one hour. After all memory is read, the troller starts a new pass at the beginning of memory.

Accelerated trolling: 101 percent

Any value greater than 100 percent invokes one-pass accelerated trolling. All memory is trolled at a rate of approximately 6000 8 kilobyte pages per second, then trolling is disabled. This mode is intended for trolling all memory quickly during off peak hours. For example, on a GS320 system with 32 processors and 128 gigabytes of memory, one-pass accelerated trolling takes approximately five minutes.

Enter the following command to display the current value of vm_troll_percent (the troll rate):

```
# /sbin/sysconfig -q vm vm_troll_percent
```

You can override the default troll rate by adding the following lines to the /etc/sysconfigtab file:

```
vm:
 vm_troll_percent=percent_rate
```

The *percent_rate* variable is the troll rate as described previously. Use the sysconfigdb command to add entries to the /etc/sysconfigtab file. See sysconfigdb(8) for more information. The new rate takes effect on the next system boot.

You can enable, disable, or change the troll rate at any time using the following command:

```
# /sbin/sysconfig -r vm vm_troll_percent=percent_rate
```

The *percent_rate* variable is the troll rate as described previously. Only the superuser (root) or a user authorized by division of privileges (DOP) can use this command. See dop(8) for more information on sharing superuser privileges.

6.2.1 Understanding the Configuration Messages

If the memory troller does not support your system, the following error is displayed on your terminal when you attempt to configure the memory troller using /sbin/sysconfig:

```
vm_configure: Memory Trolling not supported on this system.
```

Enter the following command to disable trolling:

```
#  /sbin/sysconfig -r vm vm_troll_percent=0
```

The following warning message is displayed on your terminal when the preceding command is executed:

```
vm_configure: shutting down memory troller.
[WARNING: disabling the memory troller is not recommended on
this system.]
```

This message notifies you that permanently disabling memory trolling is not recommended.

6.2.2 Configuring Accelerated Trolling

To schedule one-pass accelerated trolling at off peak hours, follow this procedure:

1. Create a shell script named /usr/local/fast_troll.sh containing the following lines:

   ```
   #!/sbin/sh

   /sbin/sysconfig -r vm vm_troll_percent=101
   ```

2. Enter the following commands to set the file owner and permissions of /usr/local/fast_troll.sh:

   ```
   #  chown root /usr/local/fast_troll.sh
   #  chmod 744 /usr/local/fast_troll.sh
   ```

3. Use the cron facility to schedule execution of the shell script as root user at the wanted time. See cron(8) for more information.

6.3 Controlling the Use of System Resources

Low trolling rates, such as the 4 percent default, have negligible impact on system performance. Processor usage for memory trolling increases as the troll rate is increased. To approximate the performance overhead, use the following procedure:

1. Log in as root or become superuser.
2. Choose a time when the system is idle and disable the memory troller.

Enter the following command to dissable the memory troller:

```
# /sbin/sysconfig -r vm vm_troll_percent=0
```

3. Enter the following command with the memory troller disabled to establish a performance baseline:

```
# vmstat 1

procs    memory         pages                          intr        cpu
r  w  u  act  free wire fault cow zero react pin pout  in  sy  cs  us  sy  id
2130 21   15K  40K 7682 104M  37M  27M  19K  22M  184  70 178 177   1   1  98
```

4. In the command output, note the system time, labeled `sy` under the `cpu` heading. Enter the following command to adjust the `vm_troll_percent` value:

```
# /sbin/sysconfig -r vm vm_troll_percent=percent_rate
```

   Repeat step 3 and note any change in the value of `sy` under the `cpu` heading.

A system time (`sy`) increase of one or less represents negligible performance cost. Repeat the procedure, adjusting the percent value of `vm_troll_percent` until the performance cost is acceptable.

For example, a GS320 system with 32 processors and 128 GB of memory will show approximately 25 percent of system time during one-pass accelerated trolling. The same system at the 4 percent default troll rate will show one percent or less system time.

6.4 Understanding Memory Troller Messages

The memory troller might produce both informational messages and error messages as described in the following sections.

6.4.1 Informational Messages

The following messages provide information about events associated with memory troller operation. These messages do not indicate a failure in the memory troller:

- If a memory page containing an uncorrectable error was located by the memory troller and the bad page will be mapped out, the following message is displayed:

  ```
  Memory Troller: bad page found (address = 0x################)
  ```

- In addition to the `bad page found...` message, machine check messages similar to the following are displayed on the system's console when the memory troller encounters a bad page:

  ```
  25-Mar-2000 17:24:25 [700] CPU machine check/exception - CPU 0
  25-Mar-2000 17:24:25 [700] CPU machine check/exception - CPU 18
  ```

These messages come from the event notification subsystem. They indicate that the machine checks resulting from the memory troller reading the bad page have been entered into the binary error log.

6.4.2 Error Messages

If any of the following error messages are displayed on the console terminal, a malfunction has occurred in the memory troller and you must contact your technical support organization.

```
VM_CONFIGURE: Memory Trolling is currently disabled on this
system
```

The memory troller has been disabled due to a fatal error.

```
adjust_troll_quantity:  null MAD pointer, disabling troller
```

A fatal internal error has occurred, the troller is disabled.

```
adjust_troll_quantity:  invalid troll_percent 0 defaulting
to 4 percent
```

The troller is active, but the troll rate is zero. The troller continues operating, but at the default troll rate. This is a serious error.

```
vm_memory_troller:  CPU # vmmt_get_mad() failed, disabling
troller
```

A fatal internal error has occurred, the troller is disabled.

```
vm_memory_troller:  MAD # invalid state [#], shutting down
```

A fatal internal error has occurred, the troller is disabled.

6.5 Memory Troller Interactions with OLAR

The Memory Troller automatically reconfigures itself when CPUs are taken off line (for removal) or placed back on line (after addition).

The Memory Troller automatically switches to another CPU if the CPU designated as the vm_primary CPU is taken off line.

The Memory Troller reconfigures itself when a new CPU comes on line (if necessary) to select a new vm_primary. If the memory troller does not need to select a new vm_primary it will not.

If there are enough CPUs available in a system, the Memory Troller avoids the following configurations:

- Resource Affinity Domains that only have memory
- Using the same processor as a vm_primary processor and for handling interrupts

Index