System Administration Guide: Oracle® Solaris Zones, Oracle Solaris 10 Containers, and Resource Management

Part No: 821–1460–02
November 2010

ORACLE®

110425@25097

Contents

Preface

This book is part of a multivolume set that covers a significant part of the Oracle Solaris operating system administration information. This book assumes that you have already installed the operating system and set up any networking software that you plan to use.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the Oracle Solaris OS: Hardware Compatibility Lists (`http://www.sun.com/bigadmin/hcl`). This document cites any implementation differences between the platform types.

In this document these x86 related terms mean the following:

- "x86" refers to the larger family of 64-bit and 32-bit x86 compatible products.
- "x64" relates specifically to 64-bit x86 compatible CPUs.
- "32-bit x86" points out specific 32-bit information about x86 based systems.

For supported systems, see the *Oracle Solaris OS: Hardware Compatibility Lists*.

About Oracle Solaris Zones

An Oracle Solaris Zone is a complete runtime environment for applications. A zone provides a virtual mapping from the application to the platform resources. Zones allow application components to be isolated from one another even though the zones share a single instance of the Oracle Solaris operating system. Resource management features permit you to allocate the quantity of resources that a workload receives.

The zone establishes boundaries for resource consumption, such as CPU. These boundaries can be expanded to adapt to changing processing requirements of the application running in the zone.

About Oracle Solaris 10 Zones

Oracle Solaris 10 Zones, also known as `solaris10` branded non-global zones, use BrandZ technology to run Oracle Solaris 10 applications on the Oracle Solaris 11 Express operating system. Applications run unmodified in the secure environment provided by the non-global zone feature. This enables you to use the Oracle Solaris 10 system to develop, test, and deploy applications. Workloads running within these branded zones can take advantage of the enhancements made to the kernel and utilize some of the innovative technologies available only on the Oracle Solaris 11 Express release.

To use this feature, see Part III, "Oracle Solaris 10 Zones."

About Using Oracle Solaris Zones on an Oracle Solaris Trusted Extensions System

For information on using zones on an Oracle Solaris Trusted Extensions system, see Chapter 16, "Managing Zones in Trusted Extensions (Tasks)," in *Oracle Solaris Trusted Extensions Configuration and Administration*. Note that only the labeled brand can be booted on an Oracle Solaris system configuration.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems that run the Oracle Solaris release. To use this book, you should have at least one to two years of UNIX system administration experience.

How the System Administration Guides Are Organized

Here is a list of the topics that are covered by the System Administration Guides.

Book Title	Topics
System Administration Guide: Basic Administration	User accounts and groups, shutting down and booting a system, and managing services
System Administration Guide: Advanced Administration	Terminals and modems, system resources (disk quotas, accounting, and crontabs), system processes, and troubleshooting Oracle Solaris software problems
System Administration Guide: Devices and File Systems	Removable media, disks and devices, file systems, and backing up and restoring data

Book Title	Topics
System Administration Guide: IP Services	TCP/IP network administration, IPv4 and IPv6 address administration, DHCP, IPsec, IKE, Oracle Solaris IP Filter, Mobile IP, and IPQoS
System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)	DNS, NIS, and LDAP naming and directory services, including transitioning from NIS to LDAP
System Administration Guide: Network Interfaces and Network Virtualization	Networking stack, NIC driver property configuration, NWAM configuration, manual network interface configuration, administration of VLANs and link aggregations, IP network multipathing (IPMP), WiFi wireless networking configuration, virtual NICs (VNICs), and network resource management
System Administration Guide: Network Services	**Moved to SMF. Use sharectl(1M) to manage NFS properties. Get SMF replacement for /etc/default/nfs**
	Web cache servers, time-related services, network file systems (NFS and Autofs), mail, SLP, and PPP
System Administration Guide: Printing	Printing topics and tasks, using services, tools, protocols, and technologies to set up and administer printing services and printers
System Administration Guide: Security Services	Auditing, device management, file security, BART, Kerberos services, PAM, Oracle Solaris Cryptographic Framework, privileges, RBAC, SASL, and Oracle Solaris Secure Shell
System Administration Guide: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management	Resource management features, which enable you to control how applications use available system resources; Oracle Solaris Zones software partitioning technology, which virtualizes operating system services to create an isolated environment for running applications; and Oracle Solaris 10 Zones, which host Oracle Solaris 10 environments running on the Oracle Solaris 11 Express kernel
Oracle Solaris SMB and Windows Interoperability Administration Guide	Oracle Solaris SMB service, which enables you to configure an Oracle Solaris system to make SMB shares available to SMB clients; Oracle Solaris SMB client, which enables you to access SMB shares; and native identity mapping service, which enables you to map user and group identities between Oracle Solaris systems and Windows systems
Oracle Solaris Trusted Extensions Configuration and Administration	System installation, configuration, and administration that is specific to Oracle Solaris Trusted Extensions
Oracle Solaris ZFS Administration Guide	ZFS storage pool and file system creation and management, snapshots, clones, backups, using access control lists (ACLs) to protect ZFS files, using ZFS on an Oracle Solaris system with zones installed, emulated volumes, and troubleshooting and data recovery

Documentation, Support, and Training

See the following web sites for additional resources:

- Documentation (`http://docs.sun.com`)
- Support (`http://www.oracle.com/us/support/systems/index.html`)
- Training (`http://education.oracle.com`) – Click the Sun link in the left navigation bar.

Oracle Welcomes Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of its documentation. If you find any errors or have any other suggestions for improvement, go to `http://docs.sun.com` and click Feedback. Indicate the title and part number of the documentation along with the chapter, section, and page number, if available. Please let us know if you want a reply.

Oracle Technology Network (`http://www.oracle.com/technetwork/index.html`) offers a range of resources related to Oracle software:

- Discuss technical problems and solutions on the Discussion Forums (`http://forums.oracle.com`).
- Get hands-on step-by-step tutorials with Oracle By Example (`http://www.oracle.com/technetwork/tutorials/index.html`).
- Download Sample Code (`http://www.oracle.com/technology/sample_code/index.html`).

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.

TABLE P–1 Typographic Conventions *(Continued)*

Typeface	Meaning	Example
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P–2 Shell Prompts

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

This chapter provides a brief overview of virtualization using the Oracle Solaris operating system.

About Virtualization

The goal of virtualization is to move from managing individual datacenter components to managing pools of resources. Successful server virtualization can lead to improved server utilization and more efficient use of server assets. Server virtualization is also important for successful server consolidation projects that maintain the isolation of separate systems.

Virtualization is driven by the need to consolidate multiple hosts and services on a single machine. Virtualization reduces costs through the sharing of hardware, infrastructure, and administration. Benefits include the following:

- Increased hardware utilization
- Greater flexibility in resource allocation
- Reduced power requirements
- Fewer management costs
- Lower cost of ownership
- Administrative and resource boundaries between applications on a system

Virtualization products offered by Oracle include the following:

- Oracle VM Server for SPARC (formerly Logical Domains), the SPARC hypervisor virtualization solution for running multiple operating system instances on a single machine simultaneously. Operating system-level virtualization features, such as zones or resource management, can be used in LDoms.
- Oracle Solaris Zones , which provide isolated execution environments within an Oracle Solaris operating system instance. This guide covers zones that run on the Oracle Solaris 11 Express release:
  - Default `ipkg` branded zones
  - Oracle Solaris 10 Zones (`solaris10` branded zones, described in Part III, "Oracle Solaris 10 Zones"
- Resource management features, which enable you to control how applications use available system resources.
- Oracle VM VirtualBox (formerly Sun VirtualBox), which allows you to run unmodified 32-bit and 64-bit operating systems as virtual machines on Intel and AMD processors, directly on your existing operating system.

- Network virtualization features used in virtualization technologies.

For an index of Oracle's virtualization products, with links to additional documentation and information, see Oracle Virtualization Technologies.

PART I

Oracle Solaris Resource Management

This part covers Oracle Solaris resource management, which enables you to control how applications use available system resources.

CHAPTER 1

Introduction to Resource Management

Oracle Solaris resource management functionality enables you to control how applications use available system resources. You can do the following:

- Allocate computing resources, such as processor time
- Monitor how the allocations are being used, then adjust the allocations as necessary
- Generate extended accounting information for analysis, billing, and capacity planning

This chapter covers the following topics.

- "Resource Management Overview" on page 31
- "When to Use Resource Management" on page 34
- "Setting Up Resource Management (Task Map)" on page 36

Resource Management Overview

Modern computing environments have to provide a flexible response to the varying workloads that are generated by different applications on a system. A *workload* is an aggregation of all processes of an application or group of applications. If resource management features are not used, the Oracle Solaris operating system responds to workload demands by adapting to new application requests dynamically. This default response generally means that all activity on the system is given equal access to resources. Oracle Solaris resource management features enable you to treat workloads individually. You can do the following:

- Restrict access to a specific resource
- Offer resources to workloads on a preferential basis
- Isolate workloads from each another

The ability to minimize cross-workload performance compromises, along with the facilities that monitor resource usage and utilization, is referred to as *resource management*. Resource management is implemented through a collection of algorithms. The algorithms handle the series of capability requests that an application presents in the course of its execution.

Resource management facilities permit you to modify the default behavior of the operating system with respect to different workloads. *Behavior* primarily refers to the set of decisions that are made by operating system algorithms when an application presents one or more resource requests to the system. You can use resource management facilities to do the following:

- Deny resources or prefer one application over another for a larger set of allocations than otherwise permitted
- Treat certain allocations collectively instead of through isolated mechanisms

The implementation of a system configuration that uses the resource management facilities can serve several purposes. You can do the following:

- Prevent an application from consuming resources indiscriminately
- Change an application's priority based on external events
- Balance resource guarantees to a set of applications against the goal of maximizing system utilization

When planning a resource-managed configuration, key requirements include the following:

- Identifying the competing workloads on the system
- Distinguishing those workloads that are not in conflict from those workloads with performance requirements that compromise the primary workloads

After you identify cooperating and conflicting workloads, you can create a resource configuration that presents the least compromise to the service goals of the business, within the limitations of the system's capabilities.

Effective resource management is enabled in the Oracle Solaris system by offering control mechanisms, notification mechanisms, and monitoring mechanisms. Many of these capabilities are provided through enhancements to existing mechanisms such as the `proc(4)` file system, processor sets, and scheduling classes. Other capabilities are specific to resource management. These capabilities are described in subsequent chapters.

Resource Classifications

A resource is any aspect of the computing system that can be manipulated with the intent to change application behavior. Thus, a resource is a capability that an application implicitly or explicitly requests. If the capability is denied or constrained, the execution of a robustly written application proceeds more slowly.

Classification of resources, as opposed to identification of resources, can be made along a number of axes. The axes could be implicitly requested as opposed to explicitly requested, time-based, such as CPU time, compared to time-independent, such as assigned CPU shares, and so forth.

Generally, scheduler-based resource management is applied to resources that the application can implicitly request. For example, to continue execution, an application implicitly requests additional CPU time. To write data to a network socket, an application implicitly requests bandwidth. Constraints can be placed on the aggregate total use of an implicitly requested resource.

Additional interfaces can be presented so that bandwidth or CPU service levels can be explicitly negotiated. Resources that are explicitly requested, such as a request for an additional thread, can be managed by constraint.

Resource Management Control Mechanisms

The three types of control mechanisms that are available in the Oracle Solaris operating system are constraints, scheduling, and partitioning.

Constraint Mechanisms

Constraints allow the administrator or application developer to set bounds on the consumption of specific resources for a workload. With known bounds, modeling resource consumption scenarios becomes a simpler process. Bounds can also be used to control ill-behaved applications that would otherwise compromise system performance or availability through unregulated resource requests.

Constraints do present complications for the application. The relationship between the application and the system can be modified to the point that the application is no longer able to function. One approach that can mitigate this risk is to gradually narrow the constraints on applications with unknown resource behavior. The resource controls feature discussed in Chapter 6, "Resource Controls (Overview)," provides a constraint mechanism. Newer applications can be written to be aware of their resource constraints, but not all application writers will choose to do this.

Scheduling Mechanisms

Scheduling refers to making a sequence of allocation decisions at specific intervals. The decision that is made is based on a predictable algorithm. An application that does not need its current allocation leaves the resource available for another application's use. Scheduling-based resource management enables full utilization of an undercommitted configuration, while providing controlled allocations in a critically committed or overcommitted scenario. The underlying algorithm defines how the term "controlled" is interpreted. In some instances, the scheduling algorithm might guarantee that all applications have some access to the resource. The fair share scheduler (FSS) described in Chapter 8, "Fair Share Scheduler (Overview)," manages application access to CPU resources in a controlled way.

Partitioning Mechanisms

Partitioning is used to bind a workload to a subset of the system's available resources. This binding guarantees that a known amount of resources is always available to the workload. The resource pools functionality that is described in Chapter 12, “Resource Pools (Overview),” enables you to limit workloads to specific subsets of the machine.

Configurations that use partitioning can avoid system-wide overcommitment. However, in avoiding this overcommitment, the ability to achieve high utilizations can be reduced. A reserved group of resources, such as processors, is not available for use by another workload when the workload bound to them is idle.

Resource Management Configuration

Portions of the resource management configuration can be placed in a network name service. This feature allows the administrator to apply resource management constraints across a collection of machines, rather than on an exclusively per-machine basis. Related work can share a common identifier, and the aggregate usage of that work can be tabulated from accounting data.

Resource management configuration and workload-oriented identifiers are described more fully in Chapter 2, “Projects and Tasks (Overview).” The extended accounting facility that links these identifiers with application resource usage is described in Chapter 4, “Extended Accounting (Overview).”

Interaction With Non-Global Zones

Resource management features can be used with zones to further refine the application environment. Interactions between these features and zones are described in applicable sections in this guide.

When to Use Resource Management

Use resource management to ensure that your applications have the required response times.

Resource management can also increase resource utilization. By categorizing and prioritizing usage, you can effectively use reserve capacity during off-peak periods, often eliminating the need for additional processing power. You can also ensure that resources are not wasted because of load variability.

Server Consolidation

Resource management is ideal for environments that consolidate a number of applications on a single server.

The cost and complexity of managing numerous machines encourages the consolidation of several applications on larger, more scalable servers. Instead of running each workload on a separate system, with full access to that system's resources, you can use resource management software to segregate workloads within the system. Resource management enables you to lower overall total cost of ownership by running and controlling several dissimilar applications on a single Oracle Solaris system.

If you are providing Internet and application services, you can use resource management to do the following:

- Host multiple web servers on a single machine. You can control the resource consumption for each web site and you can protect each site from the potential excesses of other sites.
- Prevent a faulty common gateway interface (CGI) script from exhausting CPU resources.
- Stop an incorrectly behaving application from leaking all available virtual memory.
- Ensure that one customer's applications are not affected by another customer's applications that run at the same site.
- Provide differentiated levels or classes of service on the same machine.
- Obtain accounting information for billing purposes.

Supporting a Large or Varied User Population

Use resource management features in any system that has a large, diverse user base, such as an educational institution. If you have a mix of workloads, the software can be configured to give priority to specific projects.

For example, in large brokerage firms, traders intermittently require fast access to execute a query or to perform a calculation. Other system users, however, have more consistent workloads. If you allocate a proportionately larger amount of processing power to the traders' projects, the traders have the responsiveness that they need.

Resource management is also ideal for supporting thin-client systems. These platforms provide stateless consoles with frame buffers and input devices, such as smart cards. The actual computation is done on a shared server, resulting in a timesharing type of environment. Use resource management features to isolate the users on the server. Then, a user who generates excess load does not monopolize hardware resources and significantly impact others who use the system.

Setting Up Resource Management (Task Map)

The following task map provides a high-level overview of the steps that are involved in setting up resource management on your system.

Task	Description	For Instructions
Identify the workloads on your system and categorize each workload by project.	Create project entries in either the `/etc/project` file, in the NIS map, or in the LDAP directory service.	"`project` Database" on page 41
Prioritize the workloads on your system.	Determine which applications are critical. These workloads might require preferential access to resources.	Refer to your business service goals.
Monitor real-time activity on your system.	Use performance tools to view the current resource consumption of workloads that are running on your system. You can then evaluate whether you must restrict access to a given resource or isolate particular workloads from other workloads.	`cpustat`(1M), `iostat`(1M), `mpstat`(1M), `prstat`(1M), `sar`(1), and `vmstat`(1M) man pages
Make temporary modifications to the workloads that are running on your system.	To determine which values can be altered, refer to the resource controls that are available in the Oracle Solaris system. You can update the values from the command line while the task or process is running.	"Available Resource Controls" on page 80, "Global and Local Actions on Resource Control Values" on page 86, "Temporarily Updating Resource Control Values on a Running System" on page 91 and `rctladm`(1M) and `prctl`(1) man pages.
Set resource controls and project attributes for every project entry in the `project` database or naming service project database.	Each project entry in the `/etc/project` file or the naming service project database can contain one or more resource controls or attributes. Resource controls constrain tasks and processes attached to that project. For each threshold value that is placed on a resource control, you can associate one or more actions to be taken when that value is reached. You can set resource controls by using the command-line interface.	"`project` Database" on page 41, "Local `/etc/project` File Format" on page 42, "Available Resource Controls" on page 80, "Global and Local Actions on Resource Control Values" on page 86, and Chapter 8, "Fair Share Scheduler (Overview)"
Place an upper bound on the resource consumption of physical memory by collections of processes attached to a project.	The resource cap enforcement daemon will enforce the physical memory resource cap defined for the project's `rcap.max-rss` attribute in the `/etc/project` file.	"`project` Database" on page 41 and Chapter 10, "Physical Memory Control Using the Resource Capping Daemon (Overview)"

Task	Description	For Instructions
Create resource pool configurations.	Resource pools provide a way to partition system resources, such as processors, and maintain those partitions across reboots. You can add one `project.pool` attribute to each entry in the `/etc/project` file.	"`project` Database" on page 41 and Chapter 12, "Resource Pools (Overview)"
Make the fair share scheduler (FSS) your default system scheduler.	Ensure that all user processes in either a single CPU system or a processor set belong to the same scheduling class.	"Configuring the FSS" on page 114 and `dispadmin(1M)` man page
Activate the extended accounting facility to monitor and record resource consumption on a task or process basis.	Use extended accounting data to assess current resource controls and to plan capacity requirements for future workloads. Aggregate usage on a system-wide basis can be tracked. To obtain complete usage statistics for related workloads that span more than one system, the project name can be shared across several machines.	"How to Activate Extended Accounting for Flows, Processes, Tasks, and Network Components" on page 70 and `acctadm(1M)` man page
(Optional) If you need to make additional adjustments to your configuration, you can continue to alter the values from the command line. You can alter the values while the task or process is running.	Modifications to existing tasks can be applied on a temporary basis without restarting the project. Tune the values until you are satisfied with the performance. Then, update the current values in the `/etc/project` file or in the naming service project database.	"Temporarily Updating Resource Control Values on a Running System" on page 91 and `rctladm(1M)` and `prctl(1)` man pages
(Optional) Capture extended accounting data.	Write extended accounting records for active processes and active tasks. The files that are produced can be used for planning, chargeback, and billing purposes. There is also a Practical Extraction and Report Language (Perl) interface to `libexacct` that enables you to develop customized reporting and extraction scripts.	`wracct(1M)` man page and "Perl Interface to `libexacct`" on page 65

CHAPTER 2

Projects and Tasks (Overview)

This chapter discusses the *project* and *task* facilities of Oracle Solaris resource management. Projects and tasks are used to label workloads and separate them from one another.

The following topics are covered in this chapter:

To use the projects and tasks facilities, see Chapter 3, "Administering Projects and Tasks."

Project and Task Facilities

To optimize workload response, you must first be able to identify the workloads that are running on the system you are analyzing. This information can be difficult to obtain by using either a purely process-oriented or a user-oriented method alone. In the Oracle Solaris system, you have two additional facilities that can be used to separate and identify workloads: the project and the task. The *project* provides a network-wide administrative identifier for related work. The *task* collects a group of processes into a manageable entity that represents a workload component.

The controls specified in the `project` name service database are set on the process, task, and project. Since process and task controls are inherited across `fork` and `settaskid` system calls, all processes and tasks that are created within the project inherit these controls. For information on these system calls, see the `fork(2)` and `settaskid(2)` man pages.

Based on their project or task membership, running processes can be manipulated with standard Oracle Solaris commands. The extended accounting facility can report on both process usage and task usage, and tag each record with the governing project identifier. This process enables offline workload analysis to be correlated with online monitoring. The project identifier can be shared across multiple machines through the `project` name service database. Thus, the resource consumption of related workloads that run on (or span) multiple machines can ultimately be analyzed across all of the machines.

Project Identifiers

The project identifier is an administrative identifier that is used to identify related work. The project identifier can be thought of as a workload tag equivalent to the user and group identifiers. A user or group can belong to one or more projects. These projects can be used to represent the workloads in which the user (or group of users) is allowed to participate. This membership can then be the basis of chargeback that is based on, for example, usage or initial resource allocations. Although a user must be assigned to a default project, the processes that the user launches can be associated with any of the projects of which that user is a member.

Determining a User's Default Project

To log in to the system, a user must be assigned a default project. A user is automatically a member of that default project, even if the user is not in the user or group list specified in that project.

Because each process on the system possesses project membership, an algorithm to assign a default project to the login or other initial process is necessary. The algorithm is documented in the man page `getprojent`(3C). The system follows ordered steps to determine the default project. If no default project is found, the user's login, or request to start a process, is denied.

The system sequentially follows these steps to determine a user's default project:

1. If the user has an entry with a `project` attribute defined in the `/etc/user_attr` extended user attributes database, then the value of the `project` attribute is the default project. See the `user_attr`(4) man page.
2. If a project with the name `user.`*user-id* is present in the `project` database, then that project is the default project. See the `project`(4) man page for more information.
3. If a project with the name `group.`*group-name* is present in the `project` database, where *group-name* is the name of the default group for the user, as specified in the `passwd` file, then that project is the default project. For information on the `passwd` file, see the `passwd`(4) man page.
4. If the special project `default` is present in the `project` database, then that project is the default project.

This logic is provided by the `getdefaultproj()` library function. See the `getprojent`(3PROJECT) man page for more information.

Setting User Attributes With the `useradd` and `usermod` Commands

You can use the following commands with the `-K` option and a *key*=*value* pair to set user attributes in local files:

`useradd` Set default project for user

`usermod` Modify user information

Local files can include the following:

- `/etc/group`
- `/etc/passwd`
- `/etc/project`
- `/etc/shadow`
- `/etc/user_attr`

If a network naming service such as NIS is being used to supplement the local file with additional entries, these commands cannot change information supplied by the network name service. However, the commands do verify the following against the external *naming service database*:

- Uniqueness of the user name (or role)
- Uniqueness of the user ID
- Existence of any group names specified

For more information, see the `useradd(1M)`, `usermod(1M)`, and `user_attr(4)` man pages.

`project` Database

You can store project data in a local file, in the Domain Name System (DNS), in a Network Information Service (NIS) project map, or in a Lightweight Directory Access Protocol (LDAP) directory service. The `/etc/project` file or naming service is used at login and by all requests for account management by the pluggable authentication module (PAM) to bind a user to a default project.

Note – Updates to entries in the project database, whether to the `/etc/project` file or to a representation of the database in a network naming service, are not applied to currently active projects. The updates are applied to new tasks that join the project when either the `login` or the `newtask` command is used. For more information, see the `login(1)` and `newtask(1)` man pages.

PAM Subsystem

Operations that change or set identity include logging in to the system, invoking an `rcp` or `rsh` command, using `ftp`, or using `su`. When an operation involves changing or setting an identity, a set of configurable modules is used to provide authentication, account management, credentials management, and session management.

For an overview of PAM, see Chapter 17, "Using PAM," in *System Administration Guide: Security Services*.

Naming Services Configuration

Resource management supports naming service `project` databases. The location where the `project` database is stored is defined in the `/etc/nsswitch.conf` file. By default, `files` is listed first, but the sources can be listed in any order.

```
project: files [nis] [ldap]
```

If more than one source for project information is listed, the `nsswitch.conf` file directs the routine to start searching for the information in the first source listed, and then search subsequent sources.

For more information about the `/etc/nsswitch.conf` file, see Chapter 2, "The Name Service Switch (Overview)," in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)* and `nsswitch.conf(4)`.

Local `/etc/project` File Format

If you select `files` as your `project` database source in the `nsswitch.conf` file, the login process searches the `/etc/project` file for project information. See the `projects(1)` and `project(4)` man pages for more information.

The `project` file contains a one-line entry of the following form for each project recognized by the system:

```
projname:projid:comment:user-list:group-list:attributes
```

The fields are defined as follows:

projname The name of the project. The name must be a string that consists of alphanumeric characters, underline (_) characters, hyphens (-), and periods (.). The period, which is reserved for projects with special meaning to the operating system, can only be used in the names of default projects for users. *projname* cannot contain colons (:) or newline characters.

projid — The project's unique numerical ID (PROJID) within the system. The maximum value of the *projid* field is `UID_MAX` (`2147483647`).

comment — A description of the project.

user-list — A comma-separated list of users who are allowed in the project.

Wildcards can be used in this field. An asterisk (*) allows all users to join the project. An exclamation point followed by an asterisk (!*) excludes all users from the project. An exclamation mark (!) followed by a user name excludes the specified user from the project.

group-list — A comma-separated list of groups of users who are allowed in the project.

Wildcards can be used in this field. An asterisk (*) allows all groups to join the project. An exclamation point followed by an asterisk (!*) excludes all groups from the project. An exclamation mark (!) followed by a group name excludes the specified group from the project.

attributes — A semicolon-separated list of name-value pairs, such as resource controls (see Chapter 6, "Resource Controls (Overview)"). *name* is an arbitrary string that specifies the object-related attribute, and *value* is the optional value for that attribute.

`name[=`*value*`]`

In the name-value pair, names are restricted to letters, digits, underscores, and periods. A period is conventionally used as a separator between the categories and subcategories of the resource control (rctl). The first character of an attribute name must be a letter. The name is case sensitive.

Values can be structured by using commas and parentheses to establish precedence.

A semicolon is used to separate name-value pairs. A semicolon cannot be used in a value definition. A colon is used to separate project fields. A colon cannot be used in a value definition.

Note – Routines that read this file halt if they encounter a malformed entry. Any projects that are specified after the incorrect entry are not assigned.

This example shows the default `/etc/project` file:

```
system:0::::
user.root:1::::
noproject:2::::
default:3::::
group.staff:10::::
```

This example shows the default `/etc/project` file with project entries added at the end:

```
system:0::::
user.root:1::::
noproject:2::::
default:3::::
group.staff:10::::
user.ml:2424:Lyle Personal:::
booksite:4113:Book Auction Project:ml,mp,jtd,kjh::
```

You can also add resource controls and attributes to the `/etc/project` file:

- To add resource controls for a project, see "Setting Resource Controls" on page 94.
- To define a physical memory resource cap for a project using the resource capping daemon described in `rcapd(1M)`, see "Attribute to Limit Physical Memory Usage for Projects" on page 120.
- To add a `project.pool` attribute to a project's entry, see "Creating the Configuration" on page 180.

Project Configuration for NIS

If you are using NIS, you can specify in the `/etc/nsswitch.conf` file to search the NIS project maps for projects:

```
project: nis files
```

The NIS maps, either `project.byname` or `project.bynumber`, have the same form as the `/etc/project` file:

```
projname:projid:comment:user-list:group-list:attributes
```

For more information, see Chapter 4, "Network Information Service (NIS) (Overview)," in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

Project Configuration for LDAP

If you are using LDAP, you can specify in the `/etc/nsswitch.conf` file to search the LDAP `project` database for projects:

```
project: ldap files
```

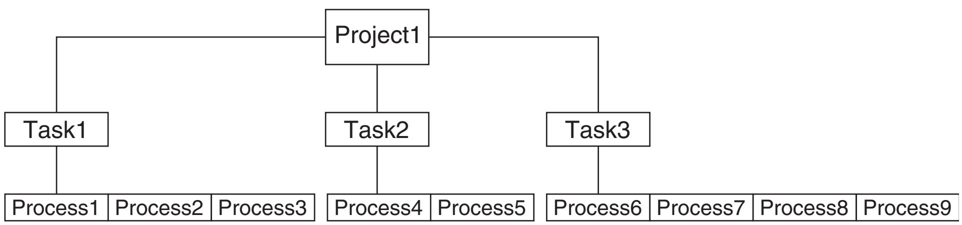
For more information about LDAP, see Chapter 8, "Introduction to LDAP Naming Services (Overview/Reference)," in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*. For more information about the schema for project entries in an LDAP database, see "Oracle Solaris Schemas" in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

Task Identifiers

Each successful login into a project creates a new *task* that contains the login process. The task is a process collective that represents a set of work over time. A task can also be viewed as a *workload component*. Each task is automatically assigned a task ID.

Each process is a member of one task, and each task is associated with one project.

FIGURE 2–1 Project and Task Tree



All operations on process groups, such as signal delivery, are also supported on tasks. You can also bind a task to a *processor set* and set a scheduling priority and class for a task, which modifies all current and subsequent processes in the task.

A task is created whenever a project is joined. The following actions, commands, and functions create tasks:

- login
- `cron`
- `newtask`
- `setproject`
- `su`

You can create a finalized task by using one of the following methods. All further attempts to create new tasks will fail.

- You can use the `newtask` command with the `-F` option.
- You can set the `task.final` attribute on a project in the `project` naming service database. All tasks created in that project by `setproject` have the `TASK_FINAL` flag.

For more information, see the login(1), newtask(1), cron(1M), su(1M), and setproject(3PROJECT) man pages.

The extended accounting facility can provide accounting data for processes. The data is aggregated at the task level.

Commands Used With Projects and Tasks

The commands that are shown in the following table provide the primary administrative interface to the project and task facilities.

Man Page Reference	Description
projects(1)	Displays project memberships for users. Lists projects from project database. Prints information on given projects. If no project names are supplied, information is displayed for all projects. Use the projects command with the -l option to print verbose output.
newtask(1)	Executes the user's default shell or specified command, placing the execution command in a new task that is owned by the specified project. newtask can also be used to change the task and the project binding for a running process. Use with the -F option to create a finalized task.
projadd(1M)	Adds a new project entry to the /etc/project file. The projadd command creates a project entry only on the local system. projadd cannot change information that is supplied by the network naming service. Can be used to edit project files other than the default file, /etc/project. Provides syntax checking for project file. Validates and edits project attributes. Supports scaled values.
projmod(1M)	Modifies information for a project on the local system. projmod cannot change information that is supplied by the network naming service. However, the command does verify the uniqueness of the project name and project ID against the external naming service. Can be used to edit project files other than the default file, /etc/project. Provides syntax checking for project file. Validates and edits project attributes. Can be used to add a new attribute, add values to an attribute, or remove an attribute. Supports scaled values. Can be used with the -A option to apply the resource control values found in the project database to the active project. Existing values that do not match the values defined in the project file are removed.
projdel(1M)	Deletes a project from the local system. projdel cannot change information that is supplied by the network naming service.

Man Page Reference	Description
`useradd`(1M)	Adds default project definitions to the local files. Use with the `-K` *key*=*value* option to add or replace user attributes.
`userdel`(1M)	Deletes a user's account from the local file.
`usermod`(1M)	Modifies a user's login information on the system. Use with the `-K` *key*=*value* option to add or replace user attributes.

CHAPTER 3

Administering Projects and Tasks

This chapter describes how to use the project and task facilities of Oracle Solaris resource management.

The following topics are covered.

- "Example Commands and Command Options" on page 50
- "Administering Projects" on page 52

For an overview of the projects and tasks facilities, see Chapter 2, "Projects and Tasks (Overview)."

Note – If you are using these facilities on an Oracle Solaris system with zones installed, only processes in the same zone will be visible through system call interfaces that take process IDs when these commands are run in a non-global zone.

Administering Projects and Tasks (Task Map)

Task	Description	For Instructions
View examples of commands and options used with projects and tasks.	Display task and project IDs, display various statistics for processes and projects that are currently running on your system.	"Example Commands and Command Options" on page 50
Define a project.	Add a project entry to the `/etc/project` file and alter values for that entry.	"How to Define a Project and View the Current Project" on page 52
Delete a project.	Remove a project entry from the `/etc/project` file.	"How to Delete a Project From the `/etc/project` File" on page 55

Task	Description	For Instructions
Validate the `project` file or project database.	Check the syntax of the `/etc/project` file or verify the uniqueness of the project name and project ID against the external naming service.	"How to Validate the Contents of the `/etc/project` File" on page 56
Obtain project membership information.	Display the current project membership of the invoking process.	"How to Obtain Project Membership Information" on page 56
Create a new task.	Create a new task in a particular project by using the `newtask` command.	"How to Create a New Task" on page 56
Associate a running process with a different task and project.	Associate a process number with a new task ID in a specified project.	"How to Move a Running Process Into a New Task" on page 57
Add and work with project attributes.	Use the project database administration commands to add, edit, validate, and remove project attributes.	"Editing and Validating Project Attributes" on page 57

Example Commands and Command Options

This section provides examples of commands and options used with projects and tasks.

Command Options Used With Projects and Tasks

`ps` Command

Use the `ps` command with the `-o` option to display task and project IDs. For example, to view the project ID, type the following:

```
# ps -o user,pid,uid,projid
USER PID   UID  PROJID
jtd  89430 124  4113
```

`id` Command

Use the `id` command with the `-p` option to print the current project ID in addition to the user and group IDs. If the *user* operand is provided, the project associated with that user's normal login is printed:

```
#  id -p
uid=124(jtd) gid=10(staff) projid=4113(booksite)
```

pgrep and pkill Commands

To match only processes with a project ID in a specific list, use the `pgrep` and `pkill` commands with the `-J` option:

```
# pgrep -J projidlist
# pkill -J projidlist
```

To match only processes with a task ID in a specific list, use the `pgrep` and `pkill` commands with the `-T` option:

```
# pgrep -T taskidlist
# pkill -T taskidlist
```

prstat Command

To display various statistics for processes and projects that are currently running on your system, use the `prstat` command with the `-J` option:

```
% prstat -J
   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
 12905 root     4472K 3640K cpu0    59    0   0:00:01 0.4% prstat/1
   829 root       43M   33M sleep   59    0   0:36:23 0.1% Xorg/1
   890 gdm        88M   26M sleep   59    0   0:22:22 0.0% gdm-simple-gree/1
   686 root     3584K 2756K sleep   59    0   0:00:34 0.0% automountd/4
     5 root        0K    0K sleep   99  -20   0:02:43 0.0% zpool-rpool/138
  9869 root       44M   17M sleep   59    0   0:02:06 0.0% poold/9
   804 root     7104K 5968K sleep   59    0   0:01:28 0.0% intrd/1
   445 root     7204K 4680K sleep   59    0   0:00:38 0.0% nscd/33
   881 gdm      7140K 5912K sleep   59    0   0:00:06 0.0% gconfd-2/1
   164 root     2572K 1648K sleep   59    0   0:00:00 0.0% pfexecd/3
   886 gdm      7092K 4920K sleep   59    0   0:00:00 0.0% bonobo-activati/2
    45 netcfg   2252K 1308K sleep   59    0   0:00:00 0.0% netcfgd/2
   142 daemon   7736K 5224K sleep   59    0   0:00:00 0.0% kcfd/3
    43 root     3036K 2020K sleep   59    0   0:00:00 0.0% dlmgmtd/5
   405 root     6824K 5400K sleep   59    0   0:00:18 0.0% hald/5
PROJID    NPROC  SWAP   RSS MEMORY      TIME  CPU PROJECT
     1        4 4728K   19M   0.9%   0:00:01 0.4% user.root
     0      111  278M  344M    17%   1:15:02 0.1% system
    10        2 1884K 9132K   0.4%   0:00:00 0.0% group.staff
     3        3 1668K 6680K   0.3%   0:00:00 0.0% default

Total: 120 processes, 733 lwps, load averages: 0.01, 0.00, 0.00
```

To display various statistics for processes and tasks that are currently running on your system, use the `prstat` command with the `-T` option:

```
% prstat -T
       PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
 12907 root     4488K 3588K cpu0    59    0   0:00:00 0.3% prstat/1
   829 root       43M   33M sleep   59    0   0:36:24 0.1% Xorg/1
   890 gdm        88M   26M sleep   59    0   0:22:22 0.0% gdm-simple-gree/1
  9869 root       44M   17M sleep   59    0   0:02:06 0.0% poold/9
     5 root        0K    0K sleep   99  -20   0:02:43 0.0% zpool-rpool/138
```

```
   445 root     7204K 4680K sleep   59    0   0:00:38 0.0% nscd/33
   881 gdm      7140K 5912K sleep   59    0   0:00:06 0.0% gconfd-2/1
   164 root     2572K 1648K sleep   59    0   0:00:00 0.0% pfexecd/3
   886 gdm      7092K 4920K sleep   59    0   0:00:00 0.0% bonobo-activati/2
    45 netcfg   2252K 1308K sleep   59    0   0:00:00 0.0% netcfgd/2
   142 daemon   7736K 5224K sleep   59    0   0:00:00 0.0% kcfd/3
    43 root     3036K 2020K sleep   59    0   0:00:00 0.0% dlmgmtd/5
   405 root     6824K 5400K sleep   59    0   0:00:18 0.0% hald/5
   311 root     3488K 2512K sleep   59    0   0:00:00 0.0% picld/4
   409 root     4356K 2768K sleep   59    0   0:00:00 0.0% hald-addon-cpuf/1
TASKID    NPROC  SWAP   RSS MEMORY      TIME  CPU PROJECT
  1401        2 2540K 8120K   0.4%   0:00:00 0.3% user.root
    94       15   84M  162M   7.9%   0:59:37 0.1% system
   561        1   37M   24M   1.2%   0:02:06 0.0% system
     0        2    0K    0K   0.0%   0:02:47 0.0% system
    46        1 4224K 5524K   0.3%   0:00:38 0.0% system
Total: 120 processes, 733 lwps, load averages: 0.01, 0.00, 0.00
```

Note – The `-J` and `-T` options cannot be used together.

Using `cron` and `su` With Projects and Tasks

`cron` Command

The `cron` command issues a `settaskid` to ensure that each `cron`, `at`, and `batch` job executes in a separate task, with the appropriate default project for the submitting user. The `at` and `batch` commands also capture the current project ID, which ensures that the project ID is restored when running an `at` job.

`su` Command

The `su` command joins the target user's default project by creating a new task, as part of simulating a login.

To switch the user's default project by using the `su` command, type the following:

```
# su user
```

Administering Projects

▼ How to Define a Project and View the Current Project

This example shows how to use the `projadd` command to add a project entry and the `projmod` command to alter that entry.

1 **Be superuser, or have the correct rights profile or authorizations.**

2 **View the default `/etc/project` file on your system by using `projects -l`.**

```
# projects -l
system
        projid : 0
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
user.root
        projid : 1
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
noproject
        projid : 2
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
default
        projid : 3
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
group.staff
        projid : 10
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
```

3 **Add a project with the name *booksite*. Assign the project to a user who is named *mark* with project ID number *4113*.**

```
# projadd -U mark -p 4113 booksite
```

4 **View the `/etc/project` file again.**

```
# projects -l
system
        projid : 0
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
user.root
        projid : 1
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
noproject
        projid : 2
        comment: ""
        users  : (none)
        groups : (none)
```

```
        attribs:
default
        projid : 3
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
group.staff
        projid : 10
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
booksite
        projid : 4113
        comment: ""
        users  : mark
        groups : (none)
        attribs:
```

5 **Add a comment that describes the project in the comment field.**

```
# projmod -c ‘Book Auction Project’ booksite
```

6 **View the changes in the `/etc/project` file.**

```
# projects -l
system
        projid : 0
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
user.root
        projid : 1
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
noproject
        projid : 2
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
default
        projid : 3
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
group.staff
        projid : 10
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
booksite
        projid : 4113
```

```
        comment: "Book Auction Project"
        users  : mark
        groups : (none)
        attribs:
```

See Also To bind projects, tasks, and processes to a pool, see "Setting Pool Attributes and Binding to a Pool" on page 174.

▼ How to Delete a Project From the `/etc/project` File

This example shows how to use the `projdel` command to delete a project.

1. **Be superuser, or have the correct rights profile or authorizations.**

2. **Remove the project *booksite* by using the `projdel` command.**

   ```
   # projdel booksite
   ```

3. **Display the `/etc/project` file.**

   ```
   # projects -l
   system
           projid : 0
           comment: ""
           users  : (none)
           groups : (none)
           attribs:
   user.root
           projid : 1
           comment: ""
           users  : (none)
           groups : (none)
           attribs:
   noproject
           projid : 2
           comment: ""
           users  : (none)
           groups : (none)
           attribs:
   default
           projid : 3
           comment: ""
           users  : (none)
           groups : (none)
           attribs:
   group.staff
           projid : 10
           comment: ""
           users  : (none)
           groups : (none)
           attribs:
   ```

4 **Log in as user *mark* and type `projects` to view the projects that are assigned to this user.**

```
# su - mark
# projects
default
```

How to Validate the Contents of the `/etc/project` File

If no editing options are given, the `projmod` command validates the contents of the `project` file.

To validate a NIS map, type the following:

```
# ypcat project | projmod -f —
```

To check the syntax of the `/etc/project` file, type the following:

```
# projmod -n
```

How to Obtain Project Membership Information

Use the `id` command with the `-p` flag to display the current project membership of the invoking process.

```
$ id -p
uid=100(mark) gid=1(other) projid=3(default)
```

▼ How to Create a New Task

1 **Log in as a member of the destination project, *booksite* in this example.**

2 **Create a new task in the *booksite* project by using the `newtask` command with the `-v` (verbose) option to obtain the system task ID.**

```
machine% newtask -v -p booksite
16
```

The execution of `newtask` creates a new task in the specified project, and places the user's default shell in this task.

3 **View the current project membership of the invoking process.**

```
machine% id -p
uid=100(mark) gid=1(other) projid=4113(booksite)
```

The process is now a member of the new project.

▼ How to Move a Running Process Into a New Task

This example shows how to associate a running process with a different task and new project. To perform this action, you must either be superuser, have the required rights profile, or be the owner of the process and be a member of the new project.

1 **Be superuser or have the required rights profile.**

Note – If you are the owner of the process or a member of the new project, you can skip this step.

---

2 **Obtain the process ID of the** *book_catalog* **process.**

```
# pgrep book_catalog
    8100
```

3 **Associate process** *8100* **with a new task ID in the** *booksite* **project.**

```
# newtask -v -p booksite -c 8100
    17
```

The `-c` option specifies that `newtask` operate on the existing named process.

4 **Confirm the task to process ID mapping.**

```
# pgrep -T 17
    8100
```

Editing and Validating Project Attributes

You can use the `projadd` and `projmod` project database administration commands to edit project attributes.

The `-K` option specifies a replacement list of attributes. Attributes are delimited by semicolons (`;`). If the `-K` option is used with the `-a` option, the attribute or attribute value is added. If the `-K` option is used with the `-r` option, the attribute or attribute value is removed. If the `-K` option is used with the `-s` option, the attribute or attribute value is substituted.

▼ How to Add Attributes and Attribute Values to Projects

Use the `projmod` command with the `-a` and `-K` options to add values to a project attribute. If the attribute does not exist, it is created.

1 **Become superuser or have the required rights profile.**

2 **Add a `task.max-lwps` resource control attribute with no values in the project *myproject*. A task entering the project has only the system value for the attribute.**

```
# projmod -a -K task.max-lwps myproject
```

3 **You can then add a value to `task.max-lwps` in the project *myproject*. The value consists of a privilege level, a threshold value, and an action associated with reaching the threshold.**

```
# projmod -a -K "task.max-lwps=(priv,100,deny)" myproject
```

4 **Because resource controls can have multiple values, you can add another value to the existing list of values by using the same options.**

```
# projmod -a -K "task.max-lwps=(priv,1000,signal=KILL)" myproject
```

The multiple values are separated by commas. The `task.max-lwps` entry now reads:

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

▼ How to Remove Attribute Values From Projects

This procedure uses the values:

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

1 **Be superuser or have the required rights profile.**

2 **To remove an attribute value from the resource control `task.max-lwps` in the project *myproject*, use the `projmod` command with the `-r` and `-K` options.**

```
# projmod -r -K "task.max-lwps=(priv,100,deny)" myproject
```

If `task.max-lwps` has multiple values, such as:

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

The first matching value would be removed. The result would then be:

```
task.max-lwps=(priv,1000,signal=KILL)
```

▼ How to Remove a Resource Control Attribute From a Project

To remove the resource control `task.max-lwps` in the project *myproject*, use the `projmod` command with the `-r` and `-K` options.

1 **Be superuser or have the required rights profile.**

2 **Remove the attribute `task.max-lwps` and all of its values from the project *myproject*:**

```
# projmod -r -K task.max-lwps myproject
```

How to Substitute Attributes and Attribute Values for Projects

To substitute a different value for the attribute `task.max-lwps` in the project *myproject*, use the `projmod` command with the `-s` and `-K` options. If the attribute does not exist, it is created.

1 **Be superuser or have the required rights profile.**

2 **Replace the current `task.max-lwps` values with the new values shown:**

```
# projmod -s -K "task.max-lwps=(priv,100,none),(priv,120,deny)" myproject
```

The result would be:

```
task.max-lwps=(priv,100,none),(priv,120,deny)
```

How to Remove the Existing Values for a Resource Control Attribute

1 **Be superuser or have the required rights profile.**

2 **To remove the current values for `task.max-lwps` from the project *myproject*, type:**

```
# projmod -s -K task.max-lwps myproject
```

CHAPTER 4

Extended Accounting (Overview)

By using the project and task facilities that are described in Chapter 2, "Projects and Tasks (Overview)," to label and separate workloads, you can monitor resource consumption by each workload. You can use the *extended accounting* subsystem to capture a detailed set of resource consumption statistics on both processes and tasks.

The following topics are covered in this chapter.

To begin using extended accounting, skip to "How to Activate Extended Accounting for Flows, Processes, Tasks, and Network Components" on page 70.

Introduction to Extended Accounting

The extended accounting subsystem labels usage records with the project for which the work was done. You can also use extended accounting, in conjunction with the Internet Protocol Quality of Service (IPQoS) flow accounting module described in Chapter 34, "Using Flow Accounting and Statistics Gathering (Tasks)," in *System Administration Guide: IP Services*, to capture network flow information on a system.

Before you can apply resource management mechanisms, you must first be able to characterize the resource consumption demands that various workloads place on a system. The extended accounting facility in the Oracle Solaris operating system provides a flexible way to record system and network resource consumption for the following:

- Tasks.
- Processes.

- Selectors provided by the IPQoS flowacct module. For more information, see ipqos(7IPP).
- Network management. See dladm(1M) and flowadm(1M).

Unlike online monitoring tools, which enable you to measure system usage in real time, extended accounting enables you to examine historical usage. You can then make assessments of capacity requirements for future workloads.

With extended accounting data available, you can develop or purchase software for resource chargeback, workload monitoring, or capacity planning.

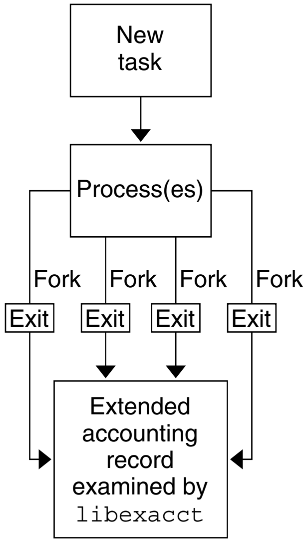
How Extended Accounting Works

The extended accounting facility in the Oracle Solaris operating system uses a versioned, extensible file format to contain accounting data. Files that use this data format can be accessed or be created by using the API provided in the included library, libexacct (see libexacct(3LIB)). These files can then be analyzed on any platform with extended accounting enabled, and their data can be used for capacity planning and chargeback.

If extended accounting is active, statistics are gathered that can be examined by the libexacct API. libexacct allows examination of the exacct files either forward or backward. The API supports third-party files that are generated by libexacct as well as those files that are created by the kernel. There is a Practical Extraction and Report Language (Perl) interface to libexacct that enables you to develop customized reporting and extraction scripts. See "Perl Interface to libexacct" on page 65.

For example, with extended accounting enabled, the task tracks the aggregate resource usage of its member processes. A task accounting record is written at task completion. Interim records on running processes and tasks can also be written. For more information on tasks, see Chapter 2, "Projects and Tasks (Overview)."

FIGURE 4–1 Task Tracking With Extended Accounting Activated



Extensible Format

The extended accounting format is substantially more extensible than the legacy system accounting software format. Extended accounting permits accounting metrics to be added and removed from the system between releases, and even during system operation.

Note – Both extended accounting and legacy system accounting software can be active on your system at the same time.

`exacct` Records and Format

Routines that allow `exacct` records to be created serve two purposes.

- To enable third-party `exacct` files to be created.
- To enable the creation of tagging records to be embedded in the kernel accounting file by using the `putacct` system call (see `getacct(2)`).

  Note – The `putacct` system call is also available from the Perl interface.

The format permits different forms of accounting records to be captured without requiring that every change be an explicit version change. Well-written applications that consume accounting data must ignore records they do not understand.

The `libexacct` library converts and produces files in the `exacct` format. This library is the *only* supported interface to `exacct` format files.

Note – The `getacct`, `putacct`, and `wracct` system calls do not apply to flows. The kernel creates flow records and writes them to the file when IPQoS flow accounting is configured.

Using Extended Accounting on an Oracle Solaris System with Zones Installed

The extended accounting subsystem collects and reports information for the entire system (including non-global zones) when run in the global zone. The global administrator or a user granted appropriate authorizations through the `zonecfg` utility can also determine resource consumption on a per-zone basis. See "Extended Accounting on a System With Zones Installed" on page 323 for more information.

Extended Accounting Configuration

The directory `/var/adm/exacct` is the standard location for placing extended accounting data. You can use the `acctadm` command to specify a different location for the process and task accounting-data files. See `acctadm(1M)` for more information.

Starting and Persistently Enabling Extended Accounting

The `acctadm` command described in `acctadm(1M)` starts extended accounting through the Oracle Solaris service management facility (SMF) service described in `smf(5)`.

The extended accounting configuration is stored in the SMF repository. The configuration is restored at boot by a service instance, one for each accounting type. Each of the extended accounting types is represented by a separate instance of the SMF service:

`svc:/system/extended-accounting:flow`
: Flow accounting

`svc:/system/extended-accounting:process`
: Process accounting

`svc:/system/extended-accounting:task`
: Task accounting

`svc:/system/extended-accounting:net`
: Network accounting

Enabling extended accounting by using acctadm(1M) causes the corresponding service instance to be enabled if not currently enabled, so that the extended accounting configuration will be restored at the next boot. Similarly, if the configuration results in accounting being disabled for a service, the service instance will be disabled. The instances are enabled or disabled by acctadm as needed.

To permanently activate extended accounting for a resource, run:

```
# acctadm -e resource_list
```

resource_list is a comma-separated list of resources or resource groups.

Records

The acctadm command appends new records to an existing /var/adm/exacct file.

Commands Used With Extended Accounting

Command Reference	Description
acctadm(1M)	Modifies various attributes of the extended accounting facility, stops and starts extended accounting, and is used to select accounting attributes to track for processes, tasks, flows and network.
wracct(1M)	Writes extended accounting records for active processes and active tasks.
lastcomm(1)	Displays previously invoked commands. lastcomm can consume either standard accounting-process data or extended-accounting process data.

For information on commands that are associated with tasks and projects, see “Example Commands and Command Options” on page 50. For information on IPQoS flow accounting, see ipqosconf(1M).

Perl Interface to libexacct

The Perl interface allows you to create Perl scripts that can read the accounting files produced by the exacct framework. You can also create Perl scripts that write exacct files.

The interface is functionally equivalent to the underlying C API. When possible, the data obtained from the underlying C API is presented as Perl data types. This feature makes accessing the data easier and it removes the need for buffer pack and unpack operations. Moreover, all memory management is performed by the Perl library.

The various project, task, and exacct-related functions are separated into groups. Each group of functions is located in a separate Perl module. Each module begins with the Sun standard Sun::Solaris:: Perl package prefix. All of the classes provided by the Perl exacct library are found under the Sun::Solaris::Exacct module.

The underlying libexacct(3LIB) library provides operations on exacct format files, catalog tags, and exacct objects. exacct objects are subdivided into two types:

- Items, which are single-data values (scalars)
- Groups, which are lists of Items

The following table summarizes each of the modules.

Module (should not contain spaces)	Description	For More Information
Sun::Solaris::Project	This module provides functions to access the project manipulation functions getprojid(2), endprojent(3PROJECT) , fgetprojent(3PROJECT), getdefaultproj(3PROJECT), getprojbyid(3PROJECT), getprojbyname(3PROJECT), getprojent(3PROJECT), getprojidbyname(3PROJECT), inproj(3PROJECT), project_walk(3PROJECT), setproject(3PROJECT) , and setprojent(3PROJECT).	Project(3PERL)
Sun::Solaris::Task	This module provides functions to access the task manipulation functions gettaskid(2) and settaskid(2).	Task(3PERL)
Sun::Solaris::Exacct	This module is the top-level exacct module. This module provides functions to access the exacct-related system calls getacct(2), putacct(2), and wracct(2). This module also provides functions to access the libexacct(3LIB) library function ea_error(3EXACCT). Constants for all of the exacct EO_*, EW_*, EXR_*, P_*, and TASK_* macros are also provided in this module.	Exacct(3PERL)
Sun::Solaris::Exacct::Catalog	This module provides object-oriented methods to access the bitfields in an exacct catalog tag. This module also provides access to the constants for the EXC_*, EXD_*, and EXD_* macros.	Exacct::Catalog(3PERL)

Module (should not contain spaces)	Description	For More Information
Sun::Solaris::Exacct:: File	This module provides object-oriented methods to access the libexacct accounting file functions ea_open(3EXACCT), ea_close(3EXACCT), ea_get_creator(3EXACCT), ea_get_hostname(3EXACCT), ea_next_object(3EXACCT), ea_previous_object(3EXACCT), and ea_write_object(3EXACCT).	Exacct::File(3PERL)
Sun::Solaris::Exacct:: Object	This module provides object-oriented methods to access an individual exacct accounting file object. An exacct object is represented as an opaque reference blessed into the appropriate Sun::Solaris::Exacct::Object subclass. This module is further subdivided into the object types Item and Group. At this level, there are methods to access the ea_match_object_catalog(3EXACCT) and ea_attach_to_object(3EXACCT) functions.	Exacct::Object(3PERL)
Sun::Solaris::Exacct:: Object::Item	This module provides object-oriented methods to access an individual exacct accounting file Item. Objects of this type inherit from Sun::Solaris::Exacct::Object.	Exacct::Object::Item(3PERL)
Sun::Solaris::Exacct:: Object::Group	This module provides object-oriented methods to access an individual exacct accounting file Group. Objects of this type inherit from Sun::Solaris::Exacct::Object. These objects provide access to the ea_attach_to_group(3EXACCT) function. The Items contained within the Group are presented as a Perl array.	Exacct::Object::Group(3PERL)
Sun::Solaris::Kstat	This module provides a Perl tied hash interface to the kstat facility. A usage example for this module can be found in /bin/kstat, which is written in Perl.	Kstat(3PERL)

For examples that show how to use the modules described in the previous table, see “Using the Perl Interface to libexacct” on page 73.

CHAPTER 5

Administering Extended Accounting (Tasks)

This chapter describes how to administer the extended accounting subsystem.

For an overview of the extending accounting subsystem, see Chapter 4, "Extended Accounting (Overview)."

Administering the Extended Accounting Facility (Task Map)

Task	Description	For Instructions
Activate the extended accounting facility.	Use extended accounting to monitor resource consumption by each project running on your system. You can use the *extended accounting* subsystem to capture historical data for tasks, processes, and flows.	"How to Activate Extended Accounting for Flows, Processes, Tasks, and Network Components" on page 70
Display extended accounting status.	Determine the status of the extended accounting facility.	"How to Display Extended Accounting Status" on page 71
View available accounting resources.	View the accounting resources available on your system.	"How to View Available Accounting Resources" on page 71
Deactivate the flow, process, task, and net accounting instances.	Turn off the extended accounting functionality.	"How to Deactivate Process, Task, Flow, and Network Management Accounting" on page 72
Use the Perl interface to the extended accounting facility.	Use the Perl interface to develop customized reporting and extraction scripts.	"Using the Perl Interface to `libexacct`" on page 73

Using Extended Accounting Functionality

Users can manage extended accounting (start accounting, stop accounting, and change accounting configuration parameters) if they have the appropriate rights profile for the accounting type to be managed:

- Extended Accounting Flow Management
- Process Management
- Task Management
- Network Management

▼ How to Activate Extended Accounting for Flows, Processes, Tasks, and Network Components

To activate the extended accounting facility for tasks, processes, flows, and network components, use the `acctadm` command. The optional final parameter to `acctadm` indicates whether the command should act on the flow, process, system task, or network accounting components of the extended accounting facility.

Note – Roles contain authorizations and privileged commands. For information on how to create the role and assign the role to a user, see Managing RBAC (Task Map) in *System Administration Guide: Security Services*.

---

1. **Be superuser or have the required rights profile.**

2. **Activate extended accounting for processes.**

```
# acctadm -e extended -f /var/adm/exacct/proc process
```

3. **Activate extended accounting for tasks.**

```
# acctadm -e extended,mstate -f /var/adm/exacct/task task
```

4. **Activate extended accounting for flows.**

```
# acctadm -e extended -f /var/adm/exacct/flow flow
```

5. **Activate extended accounting for network.**

```
# acctadm -e extended -f /var/adm/exacct/net net
```

Run `acctadm` on links and flows administered by the `dladm` and `flowadm` commands.

See Also See `acctadm(1M)` for more information.

How to Display Extended Accounting Status

Type `acctadm` without arguments to display the current status of the extended accounting facility.

```
machine% acctadm
                 Task accounting: active
            Task accounting file: /var/adm/exacct/task
          Tracked task resources: extended
        Untracked task resources: none
              Process accounting: active
         Process accounting file: /var/adm/exacct/proc
       Tracked process resources: extended
     Untracked process resources: host
                 Flow accounting: active
            Flow accounting file: /var/adm/exacct/flow
          Tracked flow resources: extended
        Untracked flow resources: none
```

In the previous example, system task accounting is active in extended mode and `mstate` mode. Process and flow accounting are active in extended mode.

Note – In the context of extended accounting, microstate (`mstate`) refers to the extended data, associated with microstate process transitions, that is available in the process usage file (see `proc(4)`). This data provides substantially more detail about the activities of the process than basic or extended records.

How to View Available Accounting Resources

Available resources can vary from system to system, and from platform to platform. Use the `acctadm` command with the `-r` option to view the accounting resource groups available on your system.

```
machine% acctadm -r
process:
extended pid,uid,gid,cpu,time,command,tty,projid,taskid,ancpid,wait-status,zone,flag,
memory,mstate     displays as one line
basic    pid,uid,gid,cpu,time,command,tty,flag
task:
extended taskid,projid,cpu,time,host,mstate,anctaskid,zone
basic    taskid,projid,cpu,time
flow:
extended
saddr,daddr,sport,dport,proto,dsfield,nbytes,npkts,action,ctime,lseen,projid,uid
basic    saddr,daddr,sport,dport,proto,nbytes,npkts,action
net:
  extended name,devname,edest,vlan_tpid,vlan_tci,sap,cpuid, \
  priority,bwlimit,curtime,ibytes,obytes,ipkts,opks,ierrpkts \
```

```
oerrpkts,saddr,daddr,sport,dport,protocol,dsfield
basic    name,devname,edest,vlan_tpid,vlan_tci,sap,cpuid, \
priority,bwlimit,curtime,ibytes,obytes,ipkts,opks,ierrpkts \
oerrpkts
```

▼ How to Deactivate Process, Task, Flow, and Network Management Accounting

To deactivate process, task, flow, and network accounting, turn off each of them individually by using the `acctadm` command with the `-x` option.

1. **Be superuser or have the required rights profile.**

2. **Turn off process accounting.**

```
# acctadm -x process
```

3. **Turn off task accounting.**

```
# acctadm -x task
```

4. **Turn off flow accounting.**

```
# acctadm -x flow
```

5. **Turn off network management accounting.**

```
# acctadm -x net
```

6. **Verify that task accounting, process accounting, flow and network accounting have been turned off.**

```
    # acctadm
            Task accounting: inactive
       Task accounting file: none
     Tracked task resources: none
   Untracked task resources: extended
         Process accounting: inactive
    Process accounting file: none
  Tracked process resources: none
Untracked process resources: extended
            Flow accounting: inactive
       Flow accounting file: none
     Tracked flow resources: none
   Untracked flow resources: extended
            Net accounting: inactive
       Net accounting file: none
     Tracked Net resources: none
   Untracked Net resources: extended
```

Using the Perl Interface to libexacct

How to Recursively Print the Contents of an exacct Object

Use the following code to recursively print the contents of an exacct object. Note that this capability is provided by the library as the Sun::Solaris::Exacct::Object::dump() function. This capability is also available through the ea_dump_object() convenience function.

```
sub dump_object
     {
             my ($obj, $indent) = @_;
             my $istr = '  ' x $indent;

             #
             # Retrieve the catalog tag.  Because we are
             # doing this in an array context, the
             # catalog tag will be returned as a (type, catalog, id)
             # triplet, where each member of the triplet will behave as
             # an integer or a string, depending on context.
             # If instead this next line provided a scalar context, e.g.
             #    my $cat  = $obj->catalog()->value();
             # then $cat would be set to the integer value of the
             # catalog tag.
             #
             my @cat = $obj->catalog()->value();

             #
             # If the object is a plain item
             #
             if ($obj->type() == &EO_ITEM) {
                     #
                     # Note: The '%s' formats provide s string context, so
                     # the components of the catalog tag will be displayed
                     # as the symbolic values. If we changed the '%s'
                     # formats to '%d', the numeric value of the components
                     # would be displayed.
                     #
                     printf("%sITEM\n%s  Catalog = %s|%s|%s\n",
                        $istr, $istr, @cat);
                     $indent++;

                     #
                     # Retrieve the value of the item.  If the item contains
                     # in turn a nested exacct object (i.e., an item or
                     # group),then the value method will return a reference
                     # to the appropriate sort of perl object
                     # (Exacct::Object::Item or Exacct::Object::Group).
                     # We could of course figure out that the item contained
                     # a nested item orgroup by examining the catalog tag in
                     # @cat and looking for a type of EXT_EXACCT_OBJECT or
                     # EXT_GROUP.
                     #
```

```
                my $val = $obj->value();
                if (ref($val)) {
                        # If it is a nested object, recurse to dump it.
                        dump_object($val, $indent);
                } else {
                        # Otherwise it is just a 'plain' value, so
                        # display it.
                        printf("%s  Value = %s\n", $istr, $val);
                }

        #
        # Otherwise we know we are dealing with a group.  Groups
        # represent contents as a perl list or array (depending on
        # context), so we can process the contents of the group
        # with a 'foreach' loop, which provides a list context.
        # In a list context the value method returns the content
        # of the group as a perl list, which is the quickest
        # mechanism, but doesn't allow the group to be modified.
        # If we wanted to modify the contents of the group we could
        # do so like this:
        #    my $grp = $obj->value();   # Returns an array reference
        #    $grp->[0] = $newitem;
        # but accessing the group elements this way is much slower.
        #
        } else {
                printf("%sGROUP\n%s  Catalog = %s|%s|%s\n",
                    $istr, $istr, @cat);
                $indent++;
                # 'foreach' provides a list context.
                foreach my $val ($obj->value()) {
                        dump_object($val, $indent);
                }
                printf("%sENDGROUP\n", $istr);
        }
}
```

How to Create a New Group Record and Write It to a File

Use this script to create a new group record and write it to a file named `/tmp/exacct`.

```
#!/usr/bin/perl

use strict;
use warnings;
use Sun::Solaris::Exacct qw(:EXACCT_ALL);
# Prototype list of catalog tags and values.
     my @items = (
             [ &EXT_STRING | &EXC_DEFAULT | &EXD_CREATOR      => "me"       ],
             [ &EXT_UINT32 | &EXC_DEFAULT | &EXD_PROC_PID     => $$         ],
             [ &EXT_UINT32 | &EXC_DEFAULT | &EXD_PROC_UID     => $<         ],
             [ &EXT_UINT32 | &EXC_DEFAULT | &EXD_PROC_GID     => $(         ],
             [ &EXT_STRING | &EXC_DEFAULT | &EXD_PROC_COMMAND => "/bin/rec" ],
     );
```

```
# Create a new group catalog object.
my $cat = ea_new_catalog(&EXT_GROUP | &EXC_DEFAULT | &EXD_NONE)

# Create a new Group object and retrieve its data array.
my $group = ea_new_group($cat);
my $ary = $group->value();

# Push the new Items onto the Group array.
foreach my $v (@items) {
        push(@$ary, ea_new_item(ea_new_catalog($v->[0]), $v->[1]));
}

# Open the exacct file, write the record & close.
my $f = ea_new_file('/tmp/exacct', &O_RDWR | &O_CREAT | &O_TRUNC)
   || die("create /tmp/exacct failed: ", ea_error_str(), "\n");
$f->write($group);
$f = undef;
```

How to Print the Contents of an exacct File

Use the following Perl script to print the contents of an exacct file.

```
#!/usr/bin/perl

     use strict;
     use warnings;
     use Sun::Solaris::Exacct qw(:EXACCT_ALL);

     die("Usage is dumpexacct <exacct file>\n") unless (@ARGV == 1);

     # Open the exact file and display the header information.
     my $ef = ea_new_file($ARGV[0], &O_RDONLY) || die(error_str());
     printf("Creator:  %s\n", $ef->creator());
     printf("Hostname: %s\n\n", $ef->hostname());

     # Dump the file contents
     while (my $obj = $ef->get()) {
             ea_dump_object($obj);
     }

     # Report any errors
     if (ea_error() != EXR_OK && ea_error() != EXR_EOF)  {
             printf("\nERROR: %s\n", ea_error_str());
             exit(1);
     }
     exit(0);
```

Example Output From Sun::Solaris::Exacct::Object->dump()

Here is example output produced by running Sun::Solaris::Exacct::Object->dump() on the file created in "How to Create a New Group Record and Write It to a File" on page 74.

```
Creator:  root
Hostname: localhost
GROUP
       Catalog = EXT_GROUP|EXC_DEFAULT|EXD_NONE
       ITEM
         Catalog = EXT_STRING|EXC_DEFAULT|EXD_CREATOR
         Value = me
       ITEM
         Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_PID
         Value = 845523
       ITEM
         Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_UID
         Value = 37845
       ITEM
         Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_GID
         Value = 10
       ITEM
         Catalog = EXT_STRING|EXC_DEFAULT|EXD_PROC_COMMAND
         Value = /bin/rec
ENDGROUP
```

CHAPTER 6

Resource Controls (Overview)

After you determine the resource consumption of workloads on your system as described in Chapter 4, "Extended Accounting (Overview)," you can place boundaries on resource usage. Boundaries prevent workloads from over-consuming resources. The *resource controls* facility is the constraint mechanism that is used for this purpose.

This chapter covers the following topics.

For information about how to administer resource controls, see Chapter 7, "Administering Resource Controls (Tasks)."

Resource Controls Concepts

In the Oracle Solaris operating system, the concept of a per-process resource limit has been extended to the task and project entities described in Chapter 2, "Projects and Tasks (Overview)." These enhancements are provided by the resource controls (rctls) facility. In addition, allocations that were set through the `/etc/system` tunables are now automatic or configured through the resource controls mechanism as well.

A resource control is identified by the prefix `zone`, `project`, `task`, or `process`. Resource controls can be observed on a system-wide basis. It is possible to update resource control values on a running system.

For a list of the standard resource controls that are available in this release, see "Available Resource Controls" on page 80 See "Resource Type Properties" on page 226 for information on available zone-wide resource controls.

Resource Limits and Resource Controls

UNIX systems have traditionally provided a resource limit facility (*rlimit*). The rlimit facility allows administrators to set one or more numerical limits on the amount of resources a process can consume. These limits include per-process CPU time used, per-process core file size, and per-process maximum heap size. *Heap size* is the amount of scratch memory that is allocated for the process data segment.

The resource controls facility provides compatibility interfaces for the resource limits facility. Existing applications that use resource limits continue to run unchanged. These applications can be observed in the same way as applications that are modified to take advantage of the resource controls facility.

Interprocess Communication and Resource Controls

Processes can communicate with each other by using one of several types of interprocess communication (IPC). IPC allows information transfer or synchronization to occur between processes. The resource controls facility provides resource controls that define the behavior of the kernel's IPC facilities. These resource controls replace the `/etc/system` tunables.

Obsolete parameters that are used to initialize the default resource control values might be included in the `/etc/system` file on this Oracle Solaris system. However, using the obsolete parameters is not recommended.

To observe which IPC objects are contributing to a project's usage, use the `ipcs` command with the `-J` option. See "How to Use `ipcs`" on page 101 to view an example display. For more information about the `ipcs` command, see `ipcs(1)`.

For information about Oracle Solaris system tuning, see the *Oracle Solaris Tunable Parameters Reference Manual*.

Resource Control Constraint Mechanisms

Resource controls provide a mechanism for the constraint of system resources. Processes, tasks, projects, and zones can be prevented from consuming amounts of specified system resources. This mechanism leads to a more manageable system by preventing over-consumption of resources.

Constraint mechanisms can be used to support capacity-planning processes. An encountered constraint can provide information about application resource needs without necessarily denying the resource to the application.

Project Attribute Mechanisms

Resource controls can also serve as a simple attribute mechanism for resource management facilities. For example, the number of CPU shares made available to a project in the fair share scheduler (FSS) scheduling class is defined by the `project.cpu-shares` resource control. Because the project is assigned a fixed number of shares by the control, the various actions associated with exceeding a control are not relevant. In this context, the current value for the `project.cpu-shares` control is considered an attribute on the specified project.

Another type of project attribute is used to regulate the resource consumption of physical memory by collections of processes attached to a project. These attributes have the prefix `rcap`, for example, `rcap.max-rss`. Like a resource control, this type of attribute is configured in the `project` database. However, while resource controls are synchronously enforced by the kernel, resource caps are asynchronously enforced at the user level by the resource cap enforcement daemon, `rcapd`. For information on `rcapd`, see Chapter 10, "Physical Memory Control Using the Resource Capping Daemon (Overview)," and `rcapd`(1M).

The `project.pool` attribute is used to specify a pool binding for a project. For more information on resource pools, see Chapter 12, "Resource Pools (Overview)."

Configuring Resource Controls and Attributes

The resource controls facility is configured through the `project` database. See Chapter 2, "Projects and Tasks (Overview)." Resource controls and other attributes are set in the final field of the `project` database entry. The values associated with each resource control are enclosed in parentheses, and appear as plain text separated by commas. The values in parentheses constitute an "action clause." Each action clause is composed of a privilege level, a threshold value, and an action that is associated with the particular threshold. Each resource control can have multiple action clauses, which are also separated by commas. The following entry defines a per-task lightweight process limit and a per-process maximum CPU time limit on a project entity. The `process.max-cpu-time` would send a process a SIGTERM after the process ran for 1 hour, and a SIGKILL if the process continued to run for a total of 1 hour and 1 minute. See Table 6–3.

```
development:101:Developers:::task.max-lwps=(privileged,10,deny);
  process.max-cpu-time=(basic,3600,signal=TERM),(priv,3660,signal=KILL)
```
typed as one line

Note – On systems that have zones enabled, zone-wide resource controls are specified in the zone configuration using a slightly different format. See "Zone Configuration Data" on page 221 for more information.

The `rctladm` command allows you to make runtime interrogations of and modifications to the resource controls facility, with *global scope*. The `prctl` command allows you to make runtime interrogations of and modifications to the resource controls facility, with *local scope*.

For more information, see "Global and Local Actions on Resource Control Values" on page 86, `rctladm(1M)` and `prctl(1)`.

Note – On a system with zones installed, you cannot use `rctladm` in a non-global zone to modify settings. You can use `rctladm` in a non-global zone to view the global logging state of each resource control.

Available Resource Controls

A list of the standard resource controls that are available in this release is shown in the following table.

The table describes the resource that is constrained by each control. The table also identifies the default units that are used by the `project` database for that resource. The default units are of two types:

- Quantities represent a limited amount.
- Indexes represent a maximum valid identifier.

Thus, `project.cpu-shares` specifies the number of shares to which the project is entitled. `process.max-file-descriptor` specifies the highest file number that can be assigned to a process by the `open(2)` system call.

TABLE 6–1 Standard Project, Task, and Process Resource Controls

Control Name	Description	Default Unit
`project.cpu-cap`	Absolute limit on the amount of CPU resources that can be consumed by a project. A value of `100` means 100% of one CPU as the `project.cpu-cap` setting. A value of `125` is 125%, because 100% corresponds to one full CPU on the system when using CPU caps.	Quantity (number of CPUs)
`project.cpu-shares`	Number of CPU shares granted to this project for use with the fair share scheduler (see `FSS(7)`).	Quantity (shares)

TABLE 6–1 Standard Project, Task, and Process Resource Controls *(Continued)*

Control Name	Description	Default Unit
`project.max-crypto-memory`	Total amount of kernel memory that can be used by `libpkcs11` for hardware crypto acceleration. Allocations for kernel buffers and session-related structures are charged against this resource control.	Size (bytes)
`project.max-locked-memory`	Total amount of physical locked memory allowed. If `priv_proc_lock_memory` is assigned to a user, consider setting this resource control as well to prevent that user from locking all memory. Note that this resource control replaced `project.max-device-locked-memory`, which has been removed.	Size (bytes)
`project.max-msg-ids`	Maximum number of message queue IDs allowed for this project.	Quantity (message queue IDs)
`project.max-port-ids`	Maximum allowable number of event ports.	Quantity (number of event ports)
project.max-processes	Maximum number of process table slots simultaneously available to this project. Note that because both normal processes and zombie processes take up process table slots, the `max-processes` control thus protects against zombies exhausting the process table. Because zombie processes do not have any LWPs by definition, the `max-lwps` control cannot protect against this possibility.	Quantity (process table slots)
`project.max-sem-ids`	Maximum number of semaphore IDs allowed for this project.	Quantity (semaphore IDs)
`project.max-shm-ids`	Maximum number of shared memory IDs allowed for this project.	Quantity (shared memory IDs)
`project.max-shm-memory`	Total amount of System V shared memory allowed for this project.	Size (bytes)
`project.max-lwps`	Maximum number of LWPs simultaneously available to this project.	Quantity (LWPs)

TABLE 6–1 Standard Project, Task, and Process Resource Controls *(Continued)*

Control Name	Description	Default Unit
`project.max-tasks`	Maximum number of tasks allowable in this project.	Quantity (number of tasks)
`project.max-contracts`	Maximum number of contracts allowed in this project.	Quantity (contracts)
`task.max-cpu-time`	Maximum CPU time that is available to this task's processes.	Time (seconds)
`task.max-lwps`	Maximum number of LWPs simultaneously available to this task's processes.	Quantity (LWPs)
task.max-processes	Maximum number of process table slots simultaneously available to this task's processes.	Quantity (process table slots)
`process.max-cpu-time`	Maximum CPU time that is available to this process.	Time (seconds)
`process.max-file-descriptor`	Maximum file descriptor index available to this process.	Index (maximum file descriptor)
`process.max-file-size`	Maximum file offset available for writing by this process.	Size (bytes)
`process.max-core-size`	Maximum size of a core file created by this process.	Size (bytes)
`process.max-data-size`	Maximum heap memory available to this process.	Size (bytes)
`process.max-stack-size`	Maximum stack memory segment available to this process.	Size (bytes)
`process.max-address-space`	Maximum amount of address space, as summed over segment sizes, that is available to this process.	Size (bytes)
`process.max-port-events`	Maximum allowable number of events per event port.	Quantity (number of events)
`process.max-sem-nsems`	Maximum number of semaphores allowed per semaphore set.	Quantity (semaphores per set)
`process.max-sem-ops`	Maximum number of semaphore operations allowed per `semop` call (value copied from the resource control at `semget()` time).	Quantity (number of operations)

TABLE 6–1 Standard Project, Task, and Process Resource Controls *(Continued)*

Control Name	Description	Default Unit
`process.max-msg-qbytes`	Maximum number of bytes of messages on a message queue (value copied from the resource control at `msgget()` time).	Size (bytes)
`process.max-msg-messages`	Maximum number of messages on a message queue (value copied from the resource control at `msgget()` time).	Quantity (number of messages)

You can display the default values for resource controls on a system that does not have any resource controls set or changed. Such a system contains no non-default entries in `/etc/system` or the `project` database. To display values, use the `prctl` command.

Zone-Wide Resource Controls

Zone-wide resource controls limit the total resource usage of all process entities within a zone. Zone-wide resource controls can also be set using global property names as described in "Setting Zone-Wide Resource Controls" on page 214 and "How to Configure the Zone" on page 238.

TABLE 6–2 Zones Resource Controls

Control Name	Description	Default Unit
`zone.cpu-cap`	Absolute limit on the amount of CPU resources that can be consumed by a non-global zone. A value of `100` means 100% of one CPU as the `project.cpu-cap` setting. A value of 125 is 125%, because 100% corresponds to one full CPU on the system when using CPU caps.	Quantity (number of CPUs)
`zone.cpu-shares`	Number of fair share scheduler (FSS) CPU shares for this zone	Quantity (shares)
`zone.max-lofi`	Maximum number of `lofi` devices that can be created by a zone. The value limits each zone's usage of the minor node namespace.	Quantity (number of `lofi` devices)

TABLE 6–2 Zones Resource Controls *(Continued)*

Control Name	Description	Default Unit
zone.max-locked-memory	Total amount of physical locked memory available to a zone. When priv_proc_lock_memory is assigned to a zone, consider setting this resource control as well to prevent that zone from locking all memory.	Size (bytes)
zone.max-lwps	Maximum number of LWPs simultaneously available to this zone	Quantity (LWPs)
zone.max-msg-ids	Maximum number of message queue IDs allowed for this zone	Quantity (message queue IDs)
zone.max-processes	Maximum number of process table slots simultaneously available to this zone. Because both normal processes and zombie processes take up process table slots, the max-processes control thus protects against zombies exhausting the process table. Because zombie processes do not have any LWPs by definition, the max-lwps control cannot protect against this possibility.	Quantity (process table slots)
zone.max-sem-ids	Maximum number of semaphore IDs allowed for this zone	Quantity (semaphore IDs)
zone.max-shm-ids	Maximum number of shared memory IDs allowed for this zone	Quantity (shared memory IDs)
zone.max-shm-memory	Total amount of System V shared memory allowed for this zone	Size (bytes)
zone.max-swap	Total amount of swap that can be consumed by user process address space mappings and tmpfs mounts for this zone.	Size (bytes)

For information on configuring zone-wide resource controls, see “Resource Type Properties” on page 226 and “How to Configure the Zone” on page 238.

Note that it is possible to apply a zone-wide resource control to the global zone. See “Using the Fair Share Scheduler on an Oracle Solaris System With Zones Installed” on page 353 for additional information.

Units Support

Global flags that identify resource control types are defined for all resource controls. The flags are used by the system to communicate basic type information to applications such as the `prctl` command. Applications use the information to determine the following:

- The unit strings that are appropriate for each resource control
- The correct scale to use when interpreting scaled values

The following global flags are available:

Global Flag	Resource Control Type String	Modifier	Scale
RCTL_GLOBAL_BYTES	bytes	B	1
		KB	2^{10}
		MB	2^{20}
		GB	2^{30}
		TB	2^{40}
		PB	2^{50}
		EB	2^{60}
RCTL_GLOBAL_SECONDS	seconds	s	1
		Ks	10^{3}
		Ms	10^{6}
		Gs	10^{9}
		Ts	10^{12}
		Ps	10^{15}
		Es	10^{18}
RCTL_GLOBAL_COUNT	count	none	1
		K	10^{3}
		M	10^{6}
		G	10^{9}
		T	10^{12}
		P	10^{15}
		E	10^{18}

Scaled values can be used with resource controls. The following example shows a scaled threshold value:

```
task.max-lwps=(priv,1K,deny)
```

Note – Unit modifiers are accepted by the `prctl`, `projadd`, and `projmod` commands. You cannot use unit modifiers in the `project` database itself.

Resource Control Values and Privilege Levels

A threshold value on a resource control constitutes an enforcement point where local actions can be triggered or global actions, such as logging, can occur.

Each threshold value on a resource control must be associated with a privilege level. The privilege level must be one of the following three types.

- Basic, which can be modified by the owner of the calling process
- Privileged, which can be modified only by privileged (superuser) callers
- System, which is fixed for the duration of the operating system instance

A resource control is guaranteed to have one system value, which is defined by the system, or resource provider. The system value represents how much of the resource the current implementation of the operating system is capable of providing.

Any number of privileged values can be defined, and only one basic value is allowed. Operations that are performed without specifying a privilege value are assigned a basic privilege by default.

The privilege level for a resource control value is defined in the privilege field of the resource control block as RCTL_BASIC, RCTL_PRIVILEGED, or RCTL_SYSTEM. See `setrctl(2)` for more information. You can use the `prctl` command to modify values that are associated with basic and privileged levels.

Global and Local Actions on Resource Control Values

There are two categories of actions on resource control values: global and local.

Global Actions on Resource Control Values

Global actions apply to resource control values for every resource control on the system. You can use the `rctladm` command described in the `rctladm(1M)` man page to perform the following actions:

- Display the global state of active system resource controls
- Set global logging actions

You can disable or enable the global logging action on resource controls. You can set the `syslog` action to a specific degree by assigning a severity level, `syslog=`*level*. The possible settings for *level* are as follows:

- `debug`
- `info`
- `notice`
- `warning`
- `err`
- `crit`
- `alert`
- `emerg`

By default, there is no global logging of resource control violations. The level `n/a` indicates resource controls on which no global action can be configured.

Local Actions on Resource Control Values

Local actions are taken on a process that attempts to exceed the control value. For each threshold value that is placed on a resource control, you can associate one or more actions. There are three types of local actions: `none`, `deny`, and `signal=`. These three actions are used as follows:

`none` No action is taken on resource requests for an amount that is greater than the threshold. This action is useful for monitoring resource usage without affecting the progress of applications. You can also enable a global message that displays when the resource control is exceeded, although the process exceeding the threshhold is not affected.

`deny` You can deny resource requests for an amount that is greater than the threshold. For example, a `task.max-lwps` resource control with action `deny` causes a `fork` system call to fail if the new process would exceed the control value. See the `fork(2)` man page.

`signal=` You can enable a global signal message action when the resource control is exceeded. A signal is sent to the process when the threshold value is exceeded. Additional signals are not sent if the process consumes additional resources. Available signals are listed in Table 6–3.

Not all of the actions can be applied to every resource control. For example, a process cannot exceed the number of CPU shares assigned to the project of which it is a member. Therefore, a deny action is not allowed on the `project.cpu-shares` resource control.

Due to implementation restrictions, the global properties of each control can restrict the range of available actions that can be set on the threshold value. (See the `rctladm(1M)` man page.) A list of available signal actions is presented in the following table. For additional information about signals, see the `signal(3HEAD)` man page.

TABLE 6–3 Signals Available to Resource Control Values

Signal	Description	Notes
SIGABRT	Terminate the process.	
SIGHUP	Send a hangup signal. Occurs when carrier drops on an open line. Signal sent to the process group that controls the terminal.	
SIGTERM	Terminate the process. Termination signal sent by software.	
SIGKILL	Terminate the process and kill the program.	
SIGSTOP	Stop the process. Job control signal.	
SIGXRES	Resource control limit exceeded. Generated by resource control facility.	
SIGXFSZ	Terminate the process. File size limit exceeded.	Available only to resource controls with the RCTL_GLOBAL_FILE_SIZE property (`process.max-file-size`). See `rctlblk_set_value(3C)` for more information.
SIGXCPU	Terminate the process. CPU time limit exceeded.	Available only to resource controls with the RCTL_GLOBAL_CPUTIME property (`process.max-cpu-time`). See `rctlblk_set_value(3C)` for more information.

Resource Control Flags and Properties

Each resource control on the system has a certain set of associated properties. This set of properties is defined as a set of flags, which are associated with all controlled instances of that resource. Global flags cannot be modified, but the flags can be retrieved by using either `rctladm` or the `getrctl` system call.

Local flags define the default behavior and configuration for a specific threshold value of that resource control on a specific process or process collective. The local flags for one threshold value do not affect the behavior of other defined threshold values for the same resource control. However, the global flags affect the behavior for every value associated with a particular control. Local flags can be modified, within the constraints supplied by their corresponding global flags, by the `prctl` command or the `setrctl` system call. See `setrctl(2)`.

For the complete list of local flags, global flags, and their definitions, see rctlblk_set_value(3C).

To determine system behavior when a threshold value for a particular resource control is reached, use `rctladm` to display the global flags for the resource control . For example, to display the values for `process.max-cpu-time`, type the following:

```
$ rctladm process.max-cpu-time
    process.max-cpu-time  syslog=off  [ lowerable no-deny cpu-time inf seconds ]
```

The global flags indicate the following.

`lowerable`	Superuser privileges are not required to lower the privileged values for this control.
`no-deny`	Even when threshold values are exceeded, access to the resource is never denied.
`cpu-time`	SIGXCPU is available to be sent when threshold values of this resource are reached.
`seconds`	The time value for the resource control.
`no-basic`	Resource control values with the privilege type `basic` cannot be set. Only privileged resource control values are allowed.
`no-signal`	A local signal action cannot be set on resource control values.
`no-syslog`	The global `syslog` message action may not be set for this resource control.
`deny`	Always deny request for resource when threshold values are exceeded.
`count`	A count (integer) value for the resource control.
`bytes`	Unit of size for the resource control.

Use the `prctl` command to display local values and actions for the resource control.

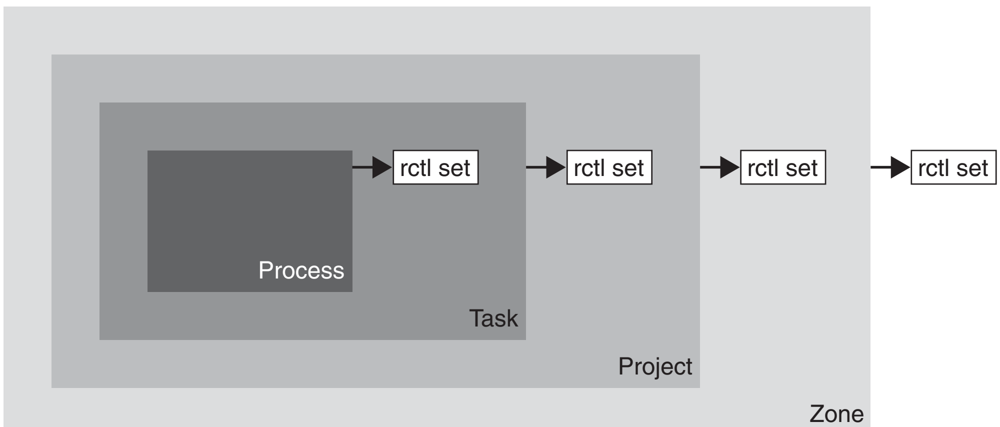
```
$ prctl -n process.max-cpu-time $$
    process 353939: -ksh
    NAME    PRIVILEGE    VALUE    FLAG   ACTION              RECIPIENT
 process.max-cpu-time
         privileged   18.4Es    inf   signal=XCPU                 -
         system       18.4Es    inf   none
```

The `max` (RCTL_LOCAL_MAXIMAL) flag is set for both threshold values, and the `inf` (RCTL_GLOBAL_INFINITE) flag is defined for this resource control. An `inf` value has an infinite quantity. The value is never enforced. Hence, as configured, both threshold quantities represent infinite values that are never exceeded.

Resource Control Enforcement

More than one resource control can exist on a resource. A resource control can exist at each containment level in the process model. If resource controls are active on the same resource at different container levels, the smallest container's control is enforced first. Thus, action is taken on `process.max-cpu-time` before `task.max-cpu-time` if both controls are encountered simultaneously.

FIGURE 6–1 Process Collectives, Container Relationships, and Their Resource Control Sets



Global Monitoring of Resource Control Events

Often, the resource consumption of processes is unknown. To get more information, try using the global resource control actions that are available with the `rctladm` command. Use `rctladm` to establish a `syslog` action on a resource control. Then, if any entity managed by that resource control encounters a threshold value, a system message is logged at the configured logging level. See Chapter 7, "Administering Resource Controls (Tasks)," and the `rctladm(1M)` man page for more information.

Applying Resource Controls

Each resource control listed in Table 6–1 can be assigned to a project at login or when `newtask`, `su`, or the other project-aware launchers `at`, `batch`, or `cron` are invoked. Each command that is initiated is launched in a separate task with the invoking user's default project. See the man pages `login(1)`, `newtask(1)`, `at(1)`, `cron(1M)`, and `su(1M)` for more information.

Updates to entries in the `project` database, whether to the `/etc/project` file or to a representation of the database in a network name service, are not applied to currently active projects. The updates are applied when a new task joins the project through login or `newtask`.

Temporarily Updating Resource Control Values on a Running System

Values changed in the `project` database only become effective for new tasks that are started in a project. However, you can use the `rctladm` and `prctl` commands to update resource controls on a running system.

Updating Logging Status

The `rctladm` command affects the global logging state of each resource control on a system-wide basis. This command can be used to view the global state and to set up the level of `syslog` logging when controls are exceeded.

Updating Resource Controls

You can view and temporarily alter resource control values and actions on a per-process, per-task, or per-project basis by using the `prctl` command. A project, task, or process ID is given as input, and the command operates on the resource control at the level where the control is defined.

Any modifications to values and actions take effect immediately. However, these modifications apply to the current process, task, or project only. The changes are not recorded in the `project` database. If the system is restarted, the modifications are lost. Permanent changes to resource controls must be made in the `project` database.

All resource control settings that can be modified in the `project` database can also be modified with the `prctl` command. Both basic and privileged values can be added or be deleted. Their actions can also be modified. By default, the basic type is assumed for all set operations, but processes and users with superuser privileges can also modify privileged resource controls. System resource controls cannot be altered.

Commands Used With Resource Controls

The commands that are used with resource controls are shown in the following table.

Command Reference	Description
ipcs(1)	Allows you to observe which IPC objects are contributing to a project's usage
prctl(1)	Allows you to make runtime interrogations of and modifications to the resource controls facility, with local scope
rctladm(1M)	Allows you to make runtime interrogations of and modifications to the resource controls facility, with global scope

The resource_controls(5) man page describes resource controls available through the project database, including units and scaling factors.

CHAPTER 7

Administering Resource Controls (Tasks)

This chapter describes how to administer the resource controls facility.

For an overview of the resource controls facility, see Chapter 6, "Resource Controls (Overview)."

Administering Resource Controls (Task Map)

Task	Description	For Instructions
Set resource controls.	Set resource controls for a project in the `/etc/project` file.	"Setting Resource Controls" on page 94
Get or revise the resource control values for active processes, tasks, or projects, with local scope.	Make runtime interrogations of and modifications to the resource controls associated with an active process, task, or project on the system.	"Using the `prctl` Command" on page 96
On a running system, view or update the global state of resource controls.	View the global logging state of each resource control on a system-wide basis. Also set up the level of `syslog` logging when controls are exceeded.	"Using `rctladm`" on page 100
Report status of active interprocess communication (IPC) facilities.	Display information about active interprocess communication (IPC) facilities. Observe which IPC objects are contributing to a project's usage.	"Using `ipcs`" on page 101

Task	Description	For Instructions
Determine whether a web server is allocated sufficient CPU capacity.	Set a global action on a resource control. This action enables you to receive notice of any entity that has a resource control value that is set too low.	"How to Determine Whether a Web Server Is Allocated Enough CPU Capacity" on page 102

Setting Resource Controls

▼ How to Set the Maximum Number of LWPs for Each Task in a Project

This procedure adds a project named `x-files` to the `/etc/project` file and sets a maximum number of LWPs for a task created in the project.

1. **Be superuser or have the required rights profile.**

2. **Use the `projadd` command with the `-K` option to create a project called `x-files`. Set the maximum number of LWPs for each task created in the project to `3`.**

```
# projadd -K 'task.max-lwps=(privileged,3,deny)' x-files
```

3. **View the entry in the `/etc/project` file by using one of the following methods:**

   - Type:

```
# projects -l
system
        projid : 0
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
.
.
.
x-files
        projid : 100
        comment: ""
        users  : (none)
        groups : (none)
        attribs: task.max-lwps=(privileged,3,deny)
```

   - Type:

```
# cat /etc/project
system:0:System:::
.
```

```
.
.
x-files:100::::task.max-lwps=(privileged,3,deny)
```

Example 7–1 Sample Session

After implementing the steps in this procedure, when superuser creates a new task in project `x-files` by joining the project with `newtask`, superuser will not be able to create more than three LWPs while running in this task. This is shown in the following annotated sample session.

```
# newtask -p x-files csh

# prctl -n task.max-lwps $$
process: 111107: csh
NAME    PRIVILEGE    VALUE    FLAG   ACTION            RECIPIENT
task.max-lwps
        usage            3
        privileged       3       -   deny                      -
        system       2.15G     max   deny                      -
# id -p
uid=0(root) gid=1(other) projid=100(x-files)

# ps -o project,taskid -p $$
 PROJECT TASKID
 x-files    73

# csh        /* creates second LWP */

# csh        /* creates third LWP */

# csh        /* cannot create more LWPs */
Vfork failed
#
```

How to Set Multiple Controls on a Project

The `/etc/project` file can contain settings for multiple resource controls for each project as well as multiple threshold values for each control. Threshold values are defined in action clauses, which are comma-separated for multiple values.

1. **Be superuser or have the required rights profile.**

2. **Use the `projmod` command with the `-s` and `-K` options to set resource controls on project `x-files`:**

```
# projmod -s -K 'task.max-lwps=(basic,10,none),(privileged,500,deny);
process.max-file-descriptor=(basic,128,deny)' x-files     one line in file
```

The following controls are set:

- A `basic` control with no action on the maximum LWPs per task.

- A privileged deny control on the maximum LWPs per task. This control causes any LWP creation that exceeds the maximum to fail, as shown in the previous example "How to Set the Maximum Number of LWPs for Each Task in a Project" on page 94.
- A limit on the maximum file descriptors per process at the basic level, which forces the failure of any open call that exceeds the maximum.

3 View the entry in the file by using one of the following methods:

- Type:

```
# projects -l
.
.
.
x-files
        projid : 100
        comment: ""
        users  : (none)
        groups : (none)
        attribs: process.max-file-descriptor=(basic,128,deny)
                 task.max-lwps=(basic,10,none),(privileged,500,deny)      one line in file
```

- Type:

```
# cat etc/project
.
.
.
x-files:100::::process.max-file-descriptor=(basic,128,deny);
task.max-lwps=(basic,10,none),(privileged,500,deny)      one line in file
```

Using the prctl Command

Use the prctl command to make runtime interrogations of and modifications to the resource controls associated with an active process, task, or project on the system. See the prctl(1) man page for more information.

▼ How to Use the prctl Command to Display Default Resource Control Values

This procedure must be used on a system on which no resource controls have been set or changed. There can be only non-default entries in the /etc/system file or in the project database.

- **Use the prctl command on any process, such as the current shell that is running.**

```
# prctl $$
process: 3320: bash
NAME    PRIVILEGE       VALUE    FLAG   ACTION                       RECIPIENT
```

```
process.max-port-events
        privileged      65.5K       -   deny                                 -
        system          2.15G     max   deny                                 -
process.max-msg-messages
        privileged      8.19K       -   deny                                 -
        system          4.29G     max   deny                                 -
process.max-msg-qbytes
        privileged      64.0KB      -   deny                                 -
        system          16.0EB    max   deny                                 -
process.max-sem-ops
        privileged        512       -   deny                                 -
        system          2.15G     max   deny                                 -
process.max-sem-nsems
        privileged        512       -   deny                                 -
        system          32.8K     max   deny                                 -
process.max-address-space
        privileged      16.0EB    max   deny                                 -
        system          16.0EB    max   deny                                 -
process.max-file-descriptor
        basic             256       -   deny                              3320
        privileged      65.5K       -   deny                                 -
        system          2.15G     max   deny                                 -
process.max-core-size
        privileged      8.00EB    max   deny                                 -
        system          8.00EB    max   deny                                 -
process.max-stack-size
        basic           10.0MB      -   deny                              3320
        privileged      32.0TB      -   deny                                 -
        system          32.0TB    max   deny                                 -
process.max-data-size
        privileged      16.0EB    max   deny                                 -
        system          16.0EB    max   deny                                 -
process.max-file-size
        privileged      8.00EB    max   deny,signal=XFSZ                     -
        system          8.00EB    max   deny                                 -
process.max-cpu-time
        privileged      18.4Es    inf   signal=XCPU                          -
        system          18.4Es    inf   none                                 -
task.max-cpu-time
        usage              0s
        system          18.4Es    inf   none                                 -
task.max-processes
        usage               2
        system          2.15G     max   deny                                 -
task.max-lwps
        usage               3
        system          2.15G     max   deny                                 -
project.max-contracts
        privileged      10.0K       -   deny                                 -
        system          2.15G     max   deny                                 -
project.max-locked-memory
        usage              0B
        system          16.0EB    max   deny                                 -
project.max-port-ids
        privileged      8.19K       -   deny                                 -
        system          65.5K     max   deny                                 -
project.max-shm-memory
        privileged       510MB      -   deny                                 -
        system          16.0EB    max   deny                                 -
```

```
project.max-shm-ids
        privileged        128       -   deny                                 -
        system          16.8M     max   deny                                 -
project.max-msg-ids
        privileged        128       -   deny                                 -
        system          16.8M     max   deny                                 -
project.max-sem-ids
        privileged        128       -   deny                                 -
        system          16.8M     max   deny                                 -
project.max-crypto-memory
        usage              0B
        privileged       510MB      -   deny                                 -
        system          16.0EB    max   deny                                 -
project.max-tasks
        usage               2
        system          2.15G     max   deny                                 -
project.max-processes
        usage               4
        system          2.15G     max   deny                                 -
project.max-lwps
        usage              11
        system          2.15G     max   deny                                 -
project.cpu-cap
        usage               0
        system          4.29G     inf   deny                                 -
project.cpu-shares
        usage               1
        privileged          1       -   none                                 -
        system          65.5K     max   none                                 -
zone.max-lofi
        usage               0
        system          18.4E     max   deny                                 -
zone.max-swap
        usage           180MB
        system          16.0EB    max   deny                                 -
zone.max-locked-memory
        usage              0B
        system          16.0EB    max   deny                                 -
zone.max-shm-memory
        system          16.0EB    max   deny                                 -
zone.max-shm-ids
        system          16.8M     max   deny                                 -
zone.max-sem-ids
        system          16.8M     max   deny                                 -
zone.max-msg-ids
        system          16.8M     max   deny                                 -
zone.max-processes
        usage              73
        system          2.15G     max   deny                                 -
zone.max-lwps
        usage             384
        system          2.15G     max   deny                                 -
zone.cpu-cap
        usage               0
        system          4.29G     inf   deny                                 -
zone.cpu-shares
        usage               1
        privileged          1       -   none                                 -
        system          65.5K     max   none
```

How to Use the prctl Command to Display Information for a Given Resource Control

- **Display the maximum file descriptor for the current shell that is running.**

```
# prctl -n process.max-file-descriptor $$
process: 110453: -sh
NAME    PRIVILEGE       VALUE    FLAG   ACTION       RECIPIENT
process.max-file-descriptor
        basic             256       -   deny            11731
        privileged      65.5K       -   deny                -
        system          2.15G     max   deny
```

How to Use prctl to Temporarily Change a Value

This example procedure uses the prctl command to temporarily add a new privileged value to deny the use of more than three LWPs per project for the x-files project. The result is comparable to the result in "How to Set the Maximum Number of LWPs for Each Task in a Project" on page 94.

1. **Be superuser or have the required rights profile.**

2. **Use newtask to join the x-files project.**

```
# newtask -p x-files
```

3. **Use the id command with the -p option to verify that the correct project has been joined.**

```
# id -p
uid=0(root) gid=1(other) projid=101(x-files)
```

4. **Add a new privileged value for project.max-lwps that limits the number of LWPs to three.**

```
# prctl -n project.max-lwps -t privileged -v 3 -e deny -i project x-files
```

5. **Verify the result.**

```
# prctl -n project.max-lwps -i project x-files
process: 111108: csh
NAME    PRIVILEGE    VALUE    FLAG   ACTION            RECIPIENT
project.max-lwps
        usage          203
        privileged    1000       -   deny                      -
        system       2.15G     max   deny                      -
```

How to Use prctl to Lower a Resource Control Value

1. **Be superuser or have the required rights profile.**

2 **Use the `prctl` command with the `-r` option to change the lowest value of the `process.max-file-descriptor` resource control.**

```
# prctl -n process.max-file-descriptor -r -v 128 $$
```

How to Use prctl to Display, Replace, and Verify the Value of a Control on a Project

1 **Be superuser or have the required rights profile.**

2 **Display the value of `project.cpu-shares` in the project *group.staff*.**

```
# prctl -n project.cpu-shares -i project group.staff
project: 2: group.staff
NAME    PRIVILEGE       VALUE    FLAG   ACTION     RECIPIENT
project.cpu-shares
        usage              1
        privileged         1        -   none               -
        system         65.5K      max   none
```

3 **Replace the current `project.cpu-shares` value `1` with the value `10`.**

```
# prctl -n project.cpu-shares -v 10 -r -i project group.staff
```

4 **Display the value of `project.cpu-shares` in the project `group.staff`.**

```
# prctl -n project.cpu-shares -i project group.staff
project: 2: group.staff
NAME    PRIVILEGE       VALUE    FLAG   ACTION     RECIPIENT
project.cpu-shares
        usage              1
        privileged         1        -   none                                   -
        system         65.5K      max   none
```

Using rctladm

How to Use rctladm

Use the `rctladm` command to make runtime interrogations of and modifications to the global state of the resource controls facility. See the `rctladm(1M)` man page for more information.

For example, you can use `rctladm` with the `-e` option to enable the global `syslog` attribute of a resource control. When the control is exceeded, notification is logged at the specified `syslog` level. To enable the global `syslog` attribute of `process.max-file-descriptor`, type the following:

```
# rctladm -e syslog process.max-file-descriptor
```

When used without arguments, the `rctladm` command displays the global flags, including the global type flag, for each resource control.

```
# rctladm
process.max-port-events     syslog=off  [ deny count ]
process.max-msg-messages    syslog=off  [ deny count ]
process.max-msg-qbytes      syslog=off  [ deny bytes ]
process.max-sem-ops         syslog=off  [ deny count ]
process.max-sem-nsems       syslog=off  [ deny count ]
process.max-address-space   syslog=off  [ lowerable deny no-signal bytes ]
process.max-file-descriptor syslog=off  [ lowerable deny count ]
process.max-core-size       syslog=off  [ lowerable deny no-signal bytes ]
process.max-stack-size      syslog=off  [ lowerable deny no-signal bytes ]
.
.
.
```

Using ipcs

How to Use ipcs

Use the `ipcs` utility to display information about active interprocess communication (IPC) facilities. See the ipcs(1) man page for more information.

You can use `ipcs` with the `-J` option to see which project's limit an IPC object is allocated against.

```
# ipcs -J
    IPC status from <running system> as of Wed Mar 26 18:53:15 PDT 2003
T         ID      KEY        MODE        OWNER    GROUP    PROJECT
Message Queues:
Shared Memory:
m       3600      0       --rw-rw-rw-   uname    staff    x-files
m        201      0       --rw-rw-rw-   uname    staff    x-files
m       1802      0       --rw-rw-rw-   uname    staff    x-files
m        503      0       --rw-rw-rw-   uname    staff    x-files
m        304      0       --rw-rw-rw-   uname    staff    x-files
m        605      0       --rw-rw-rw-   uname    staff    x-files
m          6      0       --rw-rw-rw-   uname    staff    x-files
m        107      0       --rw-rw-rw-   uname    staff    x-files
Semaphores:
s          0      0       --rw-rw-rw-   uname    staff    x-files
```

Capacity Warnings

A global action on a resource control enables you to receive notice of any entity that is tripping over a resource control value that is set too low.

For example, assume you want to determine whether a web server possesses sufficient CPUs for its typical workload. You could analyze `sar` data for idle CPU time and load average. You could also examine extended accounting data to determine the number of simultaneous processes that are running for the web server process.

However, an easier approach is to place the web server in a task. You can then set a global action, using `syslog`, to notify you whenever a task exceeds a scheduled number of LWPs appropriate for the machine's capabilities.

See the sar(1) man page for more information.

▼ How to Determine Whether a Web Server Is Allocated Enough CPU Capacity

1. **Use the `prctl` command to place a privileged (superuser-owned) resource control on the tasks that contain an `httpd` process. Limit each task's total number of LWPs to 40, and disable all local actions.**

   ```
   # prctl -n task.max-lwps -v 40 -t privileged -d all ‘pgrep httpd‘
   ```

2. **Enable a system log global action on the `task.max-lwps` resource control.**

   ```
   # rctladm -e syslog task.max-lwps
   ```

3. **Observe whether the workload trips the resource control.**

   If it does, you will see `/var/adm/messages` such as:

   ```
   Jan  8 10:15:15 testmachine unix: [ID 859581 kern.notice]
   NOTICE: privileged rctl task.max-lwps exceeded by task 19
   ```

CHAPTER 8

Fair Share Scheduler (Overview)

The analysis of workload data can indicate that a particular workload or group of workloads is monopolizing CPU resources. If these workloads are not violating resource constraints on CPU usage, you can modify the allocation policy for CPU time on the system. The fair share scheduling class described in this chapter enables you to allocate CPU time based on shares instead of the priority scheme of the timesharing (TS) scheduling class.

This chapter covers the following topics.

To begin using the fair share scheduler, see Chapter 9, "Administering the Fair Share Scheduler (Tasks)."

Introduction to the Scheduler

A fundamental job of the operating system is to arbitrate which processes get access to the system's resources. The process scheduler, which is also called the dispatcher, is the portion of the kernel that controls allocation of the CPU to processes. The scheduler supports the concept of scheduling classes. Each class defines a scheduling policy that is used to schedule processes within the class. The default scheduler in the Oracle Solaris operating system, the TS scheduler, tries to give every process relatively equal access to the available CPUs. However, you might want to specify that certain processes be given more resources than others.

You can use the *fair share scheduler* (FSS) to control the allocation of available CPU resources among workloads, based on their importance. This importance is expressed by the number of *shares* of CPU resources that you assign to each workload.

You give each project CPU shares to control the project's entitlement to CPU resources. The FSS guarantees a fair dispersion of CPU resources among projects that is based on allocated shares, independent of the number of processes that are attached to a project. The FSS achieves fairness by reducing a project's entitlement for heavy CPU usage and increasing its entitlement for light usage, in accordance with other projects.

The FSS consists of a kernel scheduling class module and class-specific versions of the `dispadmin(1M)` and `priocntl(1)` commands. Project shares used by the FSS are specified through the `project.cpu-shares` property in the `project(4)` database.

Note – If you are using the `project.cpu-shares` resource control on an Oracle Solaris system with zones installed, see "Zone Configuration Data" on page 221, "Resource Controls Used in Non-Global Zones" on page 322, and "Using the Fair Share Scheduler on an Oracle Solaris System With Zones Installed" on page 353.

CPU Share Definition

The term "share" is used to define a portion of the system's CPU resources that is allocated to a project. If you assign a greater number of CPU shares to a project, relative to other projects, the project receives more CPU resources from the fair share scheduler.

CPU shares are not equivalent to percentages of CPU resources. Shares are used to define the relative importance of workloads in relation to other workloads. When you assign CPU shares to a project, your primary concern is not the number of shares the project has. Knowing how many shares the project has in comparison with other projects is more important. You must also take into account how many of those other projects will be competing with it for CPU resources.

Note – Processes in projects with zero shares always run at the lowest system priority (0). These processes only run when projects with nonzero shares are not using CPU resources.

CPU Shares and Process State

In the Oracle Solaris system, a project workload usually consists of more than one process. From the fair share scheduler perspective, each project workload can be in either an *idle* state or an *active* state. A project is considered idle if none of its processes are using any CPU resources. This usually means that such processes are either *sleeping* (waiting for I/O completion) or stopped. A project is considered active if at least one of its processes is using CPU resources. The sum of shares of all active projects is used in calculating the portion of CPU resources to be assigned to projects.

When more projects become active, each project's CPU allocation is reduced, but the proportion between the allocations of different projects does not change.

CPU Share Versus Utilization

Share allocation is not the same as utilization. A project that is allocated 50 percent of the CPU resources might average only a 20 percent CPU use. Moreover, shares serve to limit CPU usage only when there is competition from other projects. Regardless of how low a project's allocation is, it always receives 100 percent of the processing power if it is running alone on the system. Available CPU cycles are never wasted. They are distributed between projects.

The allocation of a small share to a busy workload might slow its performance. However, the workload is not prevented from completing its work if the system is not overloaded.

CPU Share Examples

Assume you have a system with two CPUs running two parallel CPU-bound workloads called *A* and *B*, respectively. Each workload is running as a separate project. The projects have been configured so that project *A* is assigned S_A shares, and project *B* is assigned S_B shares.

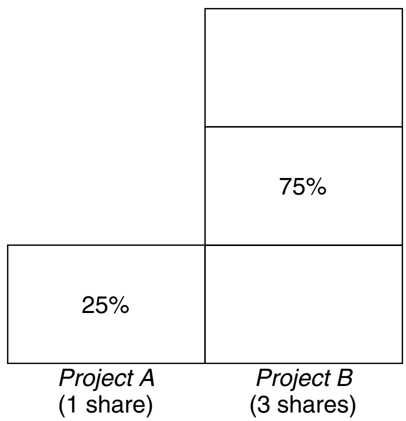
On average, under the traditional TS scheduler, each of the workloads that is running on the system would be given the same amount of CPU resources. Each workload would get 50 percent of the system's capacity.

When run under the control of the FSS scheduler with $S_A = S_B$, these projects are also given approximately the same amounts of CPU resources. However, if the projects are given different numbers of shares, their CPU resource allocations are different.

The next three examples illustrate how shares work in different configurations. These examples show that shares are only mathematically accurate for representing the usage if demand meets or exceeds available resources.

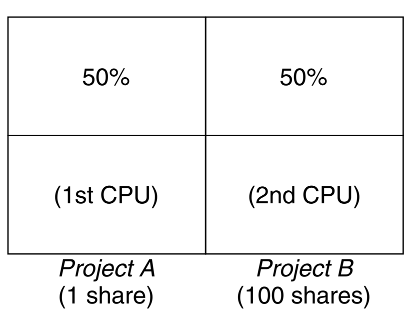
Example 1: Two CPU-Bound Processes in Each Project

If *A* and *B* each have two CPU-bound processes, and $S_A = 1$ and $S_B = 3$, then the total number of shares is 1 + 3 = 4. In this configuration, given sufficient CPU demand, projects *A* and *B* are allocated 25 percent and 75 percent of CPU resources, respectively.



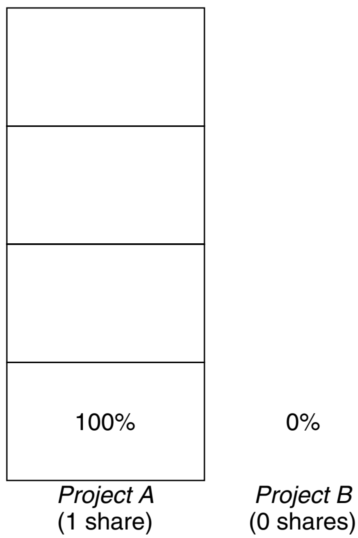
Example 2: No Competition Between Projects

If *A* and *B* have only *one* CPU-bound process each, and $S_A = 1$ and $S_B = 100$, then the total number of shares is 101. Each project cannot use more than one CPU because each project has only one running process. Because no competition exists between projects for CPU resources in this configuration, projects *A* and *B* are each allocated 50 percent of all CPU resources. In this configuration, CPU share values are irrelevant. The projects' allocations would be the same (50/50), even if both projects were assigned zero shares.



Example 3: One Project Unable to Run

If *A* and *B* have two CPU-bound processes each, and project *A* is given 1 share and project *B* is given 0 shares, then project *B* is not allocated any CPU resources and project *A* is allocated all CPU resources. Processes in *B* always run at system priority 0, so they will never be able to run because processes in project *A* always have higher priorities.



FSS Configuration

Projects and Users

Projects are the workload containers in the FSS scheduler. Groups of users who are assigned to a project are treated as single controllable blocks. Note that you can create a project with its own number of shares for an individual user.

Users can be members of multiple projects that have different numbers of shares assigned. By moving processes from one project to another project, processes can be assigned CPU resources in varying amounts.

For more information on the `project(4)` database and name services, see "`project` Database" on page 41.

CPU Shares Configuration

The configuration of CPU shares is managed by the name service as a property of the `project` database.

When the first task (or process) that is associated with a project is created through the `setproject`(3PROJECT) library function, the number of CPU shares defined as resource control `project.cpu-shares` in the `project` database is passed to the kernel. A project that does not have the `project.cpu-shares` resource control defined is assigned one share.

In the following example, this entry in the `/etc/project` file sets the number of shares for project *x-files* to *5*:

```
x-files:100::::project.cpu-shares=(privileged,5,none)
```

If you alter the number of CPU shares allocated to a project in the database when processes are already running, the number of shares for that project will not be modified at that point. The project must be restarted for the change to become effective.

If you want to temporarily change the number of shares assigned to a project without altering the project's attributes in the `project` database, use the `prctl` command. For example, to change the value of project *x-files*'s `project.cpu-shares` resource control to *3* while processes associated with that project are running, type the following:

```
# prctl -r -n project.cpu-shares -v 3 -i project x-files
```

See the `prctl`(1) man page for more information.

`-r`	Replaces the current value for the named resource control.
`-n` *name*	Specifies the name of the resource control.
`-v` *val*	Specifies the value for the resource control.
`-i` *idtype*	Specifies the ID type of the next argument.
x-files	Specifies the object of the change. In this instance, project *x-files* is the object.

Project `system` with project ID 0 includes all system daemons that are started by the boot-time initialization scripts. `system` can be viewed as a project with an unlimited number of shares. This means that `system` is always scheduled first, regardless of how many shares have been given to other projects. If you do not want the `system` project to have unlimited shares, you can specify a number of shares for this project in the `project` database.

As stated previously, processes that belong to projects with zero shares are always given zero system priority. Projects with one or more shares are running with priorities one and higher. Thus, projects with zero shares are only scheduled when CPU resources are available that are not requested by a nonzero share project.

The maximum number of shares that can be assigned to one project is 65535.

FSS and Processor Sets

The FSS can be used in conjunction with processor sets to provide more fine-grained controls over allocations of CPU resources among projects that run on each processor set than would be available with processor sets alone. The FSS scheduler treats processor sets as entirely independent partitions, with each processor set controlled independently with respect to CPU allocations.

The CPU allocations of projects running in one processor set are not affected by the CPU shares or activity of projects running in another processor set because the projects are not competing for the same resources. Projects only compete with each other if they are running within the same processor set.

The number of shares allocated to a project is system wide. Regardless of which processor set it is running on, each portion of a project is given the same amount of shares.

When processor sets are used, project CPU allocations are calculated for active projects that run within each processor set.

Project partitions that run on different processor sets might have different CPU allocations. The CPU allocation for each project partition in a processor set depends only on the allocations of other projects that run on the same processor set.

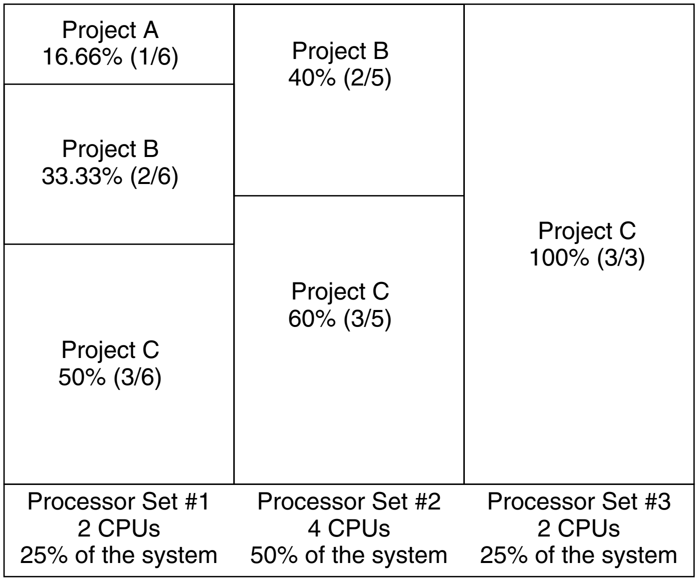
The performance and availability of applications that run within the boundaries of their processor sets are not affected by the introduction of new processor sets. The applications are also not affected by changes that are made to the share allocations of projects that run on other processor sets.

Empty processor sets (sets without processors in them) or processor sets without processes bound to them do not have any impact on the FSS scheduler behavior.

FSS and Processor Sets Examples

Assume that a server with eight CPUs is running several CPU-bound applications in projects *A*, *B*, and *C*. Project *A* is allocated one share, project *B* is allocated two shares, and project *C* is allocated three shares.

Project *A* is running only on processor set 1. Project *B* is running on processor sets 1 and 2. Project *C* is running on processor sets 1, 2, and 3. Assume that each project has enough processes to utilize all available CPU power. Thus, there is always competition for CPU resources on each processor set.



The total system-wide project CPU allocations on such a system are shown in the following table.

Project	Allocation
Project A	$4\% = (1/6 \text{ X } 2/8)_{pset1}$
Project B	$28\% = (2/6 \text{ X } 2/8)_{pset1} + (2/5 * 4/8)_{pset2}$
Project C	$67\% = (3/6 \text{ X } 2/8)_{pset1} + (3/5 \text{ X } 4/8)_{pset2} + (3/3 \text{ X } 2/8)_{pset3}$

These percentages do not match the corresponding amounts of CPU shares that are given to projects. However, within each processor set, the per-project CPU allocation ratios are proportional to their respective shares.

On the same system *without* processor sets, the distribution of CPU resources would be different, as shown in the following table.

Project	Allocation
Project A	16.66% = (1/6)
Project B	33.33% = (2/6)
Project C	50% = (3/6)

Combining FSS With Other Scheduling Classes

By default, the FSS scheduling class uses the same range of priorities (0 to 59) as the timesharing (TS), interactive (IA), and fixed priority (FX) scheduling classes. Therefore, you should avoid having processes from these scheduling classes share *the same* processor set. A mix of processes in the FSS, TS, IA, and FX classes could result in unexpected scheduling behavior.

With the use of processor sets, you can mix TS, IA, and FX with FSS in one system. However, all the processes that run on each processor set must be in *one* scheduling class, so they do not compete for the same CPUs. The FX scheduler in particular should not be used in conjunction with the FSS scheduling class unless processor sets are used. This action prevents applications in the FX class from using priorities high enough to starve applications in the FSS class.

You can mix processes in the TS and IA classes in the same processor set, or on the same system without processor sets.

The Oracle Solaris system also offers a real-time (RT) scheduler to users with superuser privileges. By default, the RT scheduling class uses system priorities in a different range (usually from 100 to 159) than FSS. Because RT and FSS are using *disjoint*, or non-overlapping, ranges of priorities, FSS can coexist with the RT scheduling class within the same processor set. However, the FSS scheduling class does not have any control over processes that run in the RT class.

For example, on a four-processor system, a single-threaded RT process can consume one entire processor if the process is CPU bound. If the system also runs FSS, regular user processes compete for the three remaining CPUs that are not being used by the RT process. Note that the RT process might not use the CPU continuously. When the RT process is idle, FSS utilizes all four processors.

You can type the following command to find out which scheduling classes the processor sets are running in and ensure that each processor set is configured to run either TS, IA, FX, or FSS processes.

```
$ ps -ef -o pset,class | grep -v CLS | sort | uniq
1 FSS
1 SYS
2 TS
2 RT
3 FX
```

Setting the Scheduling Class for the System

To set the default scheduling class for the system, see "How to Make FSS the Default Scheduler Class" on page 115, "Scheduling Class" on page 208, and `dispadmin(1M)`. To move running processes into a different scheduling class, see "Configuring the FSS" on page 114 and `priocntl(1)`.

Scheduling Class on a System with Zones Installed

Non-global zones use the default scheduling class for the system. If the system is updated with a new default scheduling class setting, non-global zones obtain the new setting when booted or rebooted.

The preferred way to use FSS in this case is to set FSS to be the system default scheduling class with the `dispadmin` command. All zones then benefit from getting a fair share of the system CPU resources. See "Scheduling Class" on page 208 for more information on scheduling class when zones are in use.

For information about moving running processes into a different scheduling class without changing the default scheduling class and rebooting, see Table 24–5 and the `priocntl(1)` man page.

Commands Used With FSS

The commands that are shown in the following table provide the primary administrative interface to the fair share scheduler.

Command Reference	Description
`priocntl(1)`	Displays or sets scheduling parameters of specified processes, moves running processes into a different scheduling class.
`ps(1)`	Lists information about running processes, identifies in which scheduling classes processor sets are running.
`dispadmin(1M)`	Sets the default scheduler for the system. Also used to examine and tune the FSS scheduler's time quantum value.
`FSS(7)`	Describes the fair share scheduler (FSS).

CHAPTER 9

Administering the Fair Share Scheduler (Tasks)

This chapter describes how to use the fair share scheduler (FSS).

For an overview of the FSS, see Chapter 8, "Fair Share Scheduler (Overview)." For information on scheduling class when zones are in use, see "Scheduling Class" on page 208.

Administering the Fair Share Scheduler (Task Map)

Task	Description	For Information
Monitor CPU usage.	Monitor the CPU usage of projects, and projects in processor sets.	"Monitoring the FSS" on page 114
Set the default scheduler class.	Make a scheduler such as the FSS the default scheduler for the system.	"How to Make FSS the Default Scheduler Class" on page 115
Move running processes from one scheduler class to a different scheduling class, such as the FSS class.	Manually move processes from one scheduling class to another scheduling class without changing the default scheduling class and rebooting.	"How to Manually Move Processes From the TS Class Into the FSS Class" on page 115
Move all running processes from all scheduling classes to a different scheduling class, such as the FSS class.	Manually move processes in all scheduling classes to another scheduling class without changing the default scheduling class and rebooting.	"How to Manually Move Processes From All User Classes Into the FSS Class" on page 115
Move a project's processes into a different scheduling class, such as the FSS class.	Manually move a project's processes from their current scheduling class to a different scheduling class.	"How to Manually Move a Project's Processes Into the FSS Class" on page 116

Task	Description	For Information
Examine and tune FSS parameters.	Tune the scheduler's time quantum value. *Time quantum* is the amount of time that a thread is allowed to run before it must relinquish the processor.	"How to Tune Scheduler Parameters" on page 116

Monitoring the FSS

You can use the `prstat` command described in the `prstat`(1M) man page to monitor CPU usage by active projects.

You can use the extended accounting data for tasks to obtain per-project statistics on the amount of CPU resources that are consumed over longer periods. See Chapter 4, "Extended Accounting (Overview)," for more information.

▼ How to Monitor System CPU Usage by Projects

● **To monitor the CPU usage of projects that run on the system, use the `prstat` command with the `-J` option.**

```
% prstat -J
```

▼ How to Monitor CPU Usage by Projects in Processor Sets

● **To monitor the CPU usage of projects on a list of processor sets, type:**

```
% prstat -J -C pset-list
```

where *pset-list* is a list of processor set IDs that are separated by commas.

Configuring the FSS

The same commands that you use with other scheduling classes in the Oracle Solaris system can be used with FSS. You can set the scheduler class, configure the scheduler's tunable parameters, and configure the properties of individual processes.

Note that you can use `svcadm restart` to restart the scheduler service. See `svcadm`(1M) for more information.

▼ How to Make FSS the Default Scheduler Class

The FSS must be the default scheduler on your system to have CPU shares assignment take effect.

Using a combination of the `priocntl` and `dispadmin` commands ensures that the FSS becomes the default scheduler immediately and also after reboot.

1. **Be superuser or have the required rights profile.**

2. **Set the default scheduler for the system to be the FSS.**

   ```
   # dispadmin -d FSS
   ```

   This change takes effect on the next reboot. After reboot, every process on the system runs in the FSS scheduling class.

3. **Make this configuration take effect immediately, without rebooting.**

   ```
   # priocntl -s -c FSS -i all
   ```

▼ How to Manually Move Processes From the TS Class Into the FSS Class

You can manually move processes from one scheduling class to another scheduling class without changing the default scheduling class and rebooting. This procedure shows how to manually move processes from the TS scheduling class into the FSS scheduling class.

1. **Be superuser or have the required rights profile.**

2. **Move the `init` process (pid 1) into the FSS scheduling class.**

   ```
   # priocntl -s -c FSS -i pid 1
   ```

3. **Move all processes from the TS scheduling class into the FSS scheduling class.**

   ```
   # priocntl -s -c FSS -i class TS
   ```

Note – All processes again run in the TS scheduling class after reboot.

▼ How to Manually Move Processes From All User Classes Into the FSS Class

You might be using a default class other than TS. For example, your system might be running a window environment that uses the IA class by default. You can manually move all processes into the FSS scheduling class without changing the default scheduling class and rebooting.

1 **Be superuser or have the required rights profile.**

2 **Move the `init` process (pid 1) into the FSS scheduling class.**

```
# priocntl -s -c FSS -i pid 1
```

3 **Move all processes from their current scheduling classes into the FSS scheduling class.**

```
# priocntl -s -c FSS -i all
```

Note – All processes again run in the default scheduling class after reboot.

▼ How to Manually Move a Project's Processes Into the FSS Class

You can manually move a project's processes from their current scheduling class to the FSS scheduling class.

1 **Be superuser or have the required rights profile.**

2 **Move processes that run in project ID *10* to the FSS scheduling class.**

```
# priocntl -s -c FSS -i projid 10
```

The project's processes again run in the default scheduling class after reboot.

How to Tune Scheduler Parameters

You can use the `dispadmin` command to display or change process scheduler parameters while the system is running. For example, you can use `dispadmin` to examine and tune the FSS scheduler's time quantum value. *Time quantum* is the amount of time that a thread is allowed to run before it must relinquish the processor.

To display the current time quantum for the FSS scheduler while the system is running, type:

```
$ dispadmin -c FSS -g
#
# Fair Share Scheduler Configuration
#
RES=1000
#
# Time Quantum
#
QUANTUM=110
```

When you use the `-g` option, you can also use the `-r` option to specify the resolution that is used for printing time quantum values. If no resolution is specified, time quantum values are displayed in milliseconds by default.

```
$ dispadmin -c FSS -g -r 100
#
# Fair Share Scheduler Configuration
#
RES=100
#
# Time Quantum
#
QUANTUM=11
```

To set scheduling parameters for the FSS scheduling class, use `dispadmin -s`. The values in *file* must be in the format output by the `-g` option. These values overwrite the current values in the kernel. Type the following:

```
$ dispadmin -c FSS -s file
```

CHAPTER 10

Physical Memory Control Using the Resource Capping Daemon (Overview)

The resource capping daemon `rcapd` enables you to regulate physical memory consumption by processes running in projects that have resource caps defined. If you are running zones on your system, you can use `rcapd` from the global zone to regulate physical memory consumption in non-global zones. See Chapter 17, "Planning and Configuring Non-Global Zones (Tasks)."

The following topics are covered in this chapter.

For procedures using the `rcapd` feature, see Chapter 11, "Administering the Resource Capping Daemon (Tasks)."

Introduction to the Resource Capping Daemon

A resource *cap* is an upper bound placed on the consumption of a resource, such as physical memory. Per-project physical memory caps are supported.

The resource capping daemon and its associated utilities provide mechanisms for physical memory resource cap enforcement and administration.

Like the resource control, the resource cap can be defined by using attributes of project entries in the `project` database. However, while resource controls are synchronously enforced by the kernel, resource caps are asynchronously enforced at the user level by the resource capping daemon. With asynchronous enforcement, a small delay occurs as a result of the sampling interval used by the daemon.

For information about rcapd, see the rcapd(1M) man page. For information about projects and the project database, see Chapter 2, "Projects and Tasks (Overview)," and the project(4) man page. For information about resource controls, see Chapter 6, "Resource Controls (Overview)."

How Resource Capping Works

The daemon repeatedly samples the resource utilization of projects that have physical memory caps. The sampling interval used by the daemon is specified by the administrator. See "Determining Sample Intervals" on page 125 for additional information. When the system's physical memory utilization exceeds the threshold for cap enforcement, and other conditions are met, the daemon takes action to reduce the resource consumption of projects with memory caps to levels at or below the caps.

The virtual memory system divides physical memory into segments known as pages. Pages are the fundamental unit of physical memory in the Oracle Solaris memory management subsystem. To read data from a file into memory, the virtual memory system reads in one page at a time, or *pages in* a file. To reduce resource consumption, the daemon can *page out*, or relocate, infrequently used pages to a swap device, which is an area outside of physical memory.

The daemon manages physical memory by regulating the size of a project workload's resident set relative to the size of its working set. The resident set is the set of pages that are resident in physical memory. The working set is the set of pages that the workload actively uses during its processing cycle. The working set changes over time, depending on the process's mode of operation and the type of data being processed. Ideally, every workload has access to enough physical memory to enable its working set to remain resident. However, the working set can also include the use of secondary disk storage to hold the memory that does not fit in physical memory.

Only one instance of rcapd can run at any given time.

Attribute to Limit Physical Memory Usage for Projects

To define a physical memory resource cap for a project, establish a resident set size (RSS) cap by adding this attribute to the project database entry:

rcap.max-rss
: The total amount of physical memory, in bytes, that is available to processes in the project.

For example, the following line in the /etc/project file sets an RSS cap of 10 gigabytes for a project named db.

```
db:100::db,root::rcap.max-rss=10737418240
```

Note – The system might round the specified cap value to a page size.

You can also use the `projmod` command to set the `rcap.max-rss` attribute in the `/etc/project` file.

For more information, see Setting the Resident Set Size Cap.

rcapd Configuration

You use the `rcapadm` command to configure the resource capping daemon. You can perform the following actions:

- Set the threshold value for cap enforcement
- Set intervals for the operations performed by `rcapd`
- Enable or disable resource capping
- Display the current status of the configured resource capping daemon

To configure the daemon, you must have superuser privileges or have the required rights profile.

Configuration changes can be incorporated into `rcapd` according to the configuration interval (see "rcapd Operation Intervals" on page 124) or on demand by sending a `SIGHUP` (see the `kill`(1) man page).

If used without arguments, `rcapadm` displays the current status of the resource capping daemon if it has been configured.

The following subsections discuss cap enforcement, cap values, and `rcapd` operation intervals.

Using the Resource Capping Daemon on a System With Zones Installed

You can control resident set size (RSS) usage of a zone by setting the `capped-memory` resource when you configure the zone. For more information, see "Physical Memory Control and the capped-memory Resource" on page 209. You can run `rcapd` *in* a zone, including the global zone, to enforce memory caps on projects in that zone.

You can set a temporary cap for the maximum amount of memory that can be consumed by a specified zone, until the next reboot. See "How to Specify a Temporary Resource Cap for a Zone" on page 131.

If you are using `rcapd` on a zone to regulate physical memory consumption by processes running in projects that have resource caps defined, you must configure the daemon in those zones.

When choosing memory caps for applications in different zones, you generally do not have to consider that the applications reside in different zones. The exception is per-zone services. Per-zone services consume memory. This memory consumption must be considered when determining the amount of physical memory for a system, as well as memory caps.

Note – You cannot run `rcapd` in an `lx` branded zone. However, you can use the daemon from the global zone to cap memory in the branded zone.

Memory Cap Enforcement Threshold

The *memory cap enforcement threshold* is the percentage of physical memory utilization on the system that triggers cap enforcement. When the system exceeds this utilization, caps are enforced. The physical memory used by applications and the kernel is included in this percentage. The percentage of utilization determines the way in which memory caps are enforced.

To enforce caps, memory can be paged out from project workloads.

- Memory can be paged out to reduce the size of the portion of memory that is over its cap for a given workload.
- Memory can be paged out to reduce the proportion of physical memory used that is over the memory cap enforcement threshold on the system.

A workload is permitted to use physical memory up to its cap. A workload can use additional memory as long as the system's memory utilization stays below the memory cap enforcement threshold.

To set the value for cap enforcement, see "How to Set the Memory Cap Enforcement Threshold" on page 129.

Determining Cap Values

If a project cap is set too low, there might not be enough memory for the workload to proceed effectively under normal conditions. The paging that occurs because the workload requires more memory has a negative effect on system performance.

Projects that have caps set too high can consume available physical memory before their caps are exceeded. In this case, physical memory is effectively managed by the kernel and not by `rcapd`.

In determining caps on projects, consider these factors.

Impact on I/O system
: The daemon can attempt to reduce a project workload's physical memory usage whenever the sampled usage exceeds the project's cap. During cap enforcement, the swap devices and other devices that contain files that the workload has mapped are used. The performance of the swap devices is a critical factor in determining the performance of a workload that routinely exceeds its cap. The execution of the workload is similar to running it on a machine with the same amount of physical memory as the workload's cap.

Impact on CPU usage
: The daemon's CPU usage varies with the number of processes in the project workloads it is capping and the sizes of the workloads' address spaces.

  A small portion of the daemon's CPU time is spent sampling the usage of each workload. Adding processes to workloads increases the time spent sampling usage.

  Another portion of the daemon's CPU time is spent enforcing caps when they are exceeded. The time spent is proportional to the amount of virtual memory involved. CPU time spent increases or decreases in response to corresponding changes in the total size of a workload's address space. This information is reported in the `vm` column of `rcapstat` output. See "Monitoring Resource Utilization With `rcapstat`" on page 125 and the `rcapstat(1)` man page for more information.

Reporting on shared memory
: The `rcapd` daemon reports the RSS of pages of memory that are shared with other processes or mapped multiple times within the same process as a reasonably accurate estimate. If processes in different projects share the same memory, then that memory will be counted towards the RSS total for all projects sharing the memory.

  The estimate is usable with workloads such as databases, which utilize shared memory extensively. For database workloads, you can also sample a project's regular usage to determine a suitable initial cap value by using output from the `-J` or `-Z` options of the `prstat` command. For more information, see the prstat(1M) man page.

rcapd Operation Intervals

You can tune the intervals for the periodic operations performed by rcapd.

All intervals are specified in seconds. The rcapd operations and their default interval values are described in the following table.

Operation	Default Interval Value in Seconds	Description
scan	15	Number of seconds between scans for processes that have joined or left a project workload. Minimum value is 1 second.
sample	5	Number of seconds between samplings of resident set size and subsequent cap enforcements. Minimum value is 1 second.
report	5	Number of seconds between updates to paging statistics. If set to 0, statistics are not updated, and output from rcapstat is not current.
config	60	Number of seconds between reconfigurations. In a reconfiguration event, rcapadm reads the configuration file for updates, and scans the project database for new or revised project caps. Sending a SIGHUP to rcapd causes an immediate reconfiguration.

To tune intervals, see "How to Set Operation Intervals" on page 129.

Determining rcapd Scan Intervals

The scan interval controls how often rcapd looks for new processes. On systems with many processes running, the scan through the list takes more time, so it might be preferable to lengthen the interval in order to reduce the overall CPU time spent. However, the scan interval also represents the minimum amount of time that a process must exist to be attributed to a capped workload. If there are workloads that run many short-lived processes, rcapd might not attribute the processes to a workload if the scan interval is lengthened.

Determining Sample Intervals

The sample interval configured with `rcapadm` is the shortest amount of time `rcapd` waits between sampling a workload's usage and enforcing the cap if it is exceeded. If you reduce this interval, `rcapd` will, under most conditions, enforce caps more frequently, possibly resulting in increased I/O due to paging. However, a shorter sample interval can also lessen the impact that a sudden increase in a particular workload's physical memory usage might have on other workloads. The window between samplings, in which the workload can consume memory unhindered and possibly take memory from other capped workloads, is narrowed.

If the sample interval specified to `rcapstat` is shorter than the interval specified to `rcapd` with `rcapadm`, the output for some intervals can be zero. This situation occurs because `rcapd` does not update statistics more frequently than the interval specified with `rcapadm`. The interval specified with `rcapadm` is independent of the sampling interval used by `rcapstat`.

Monitoring Resource Utilization With `rcapstat`

Use `rcapstat` to monitor the resource utilization of capped projects. To view an example `rcapstat` report, see "Producing Reports With rcapstat" on page 131.

You can set the sampling interval for the report and specify the number of times that statistics are repeated.

interval Specifies the sampling interval in seconds. The default interval is 5 seconds.

count Specifies the number of times that the statistics are repeated. By default, `rcapstat` reports statistics until a termination signal is received or until the `rcapd` process exits.

The paging statistics in the first report issued by `rcapstat` show the activity since the daemon was started. Subsequent reports reflect the activity since the last report was issued.

The following table defines the column headings in an `rcapstat` report.

`rcapstat` Column Headings	Description
`id`	The project ID of the capped project.
`project`	The project name.
`nproc`	The number of processes in the project.
`vm`	The total amount of virtual memory size used by processes in the project, including all mapped files and devices, in kilobytes (K), megabytes (M), or gigabytes (G).

rcapstat Column Headings	Description
rss	The estimated amount of the total resident set size (RSS) of the processes in the project, in kilobytes (K), megabytes (M), or gigabytes (G), not accounting for pages that are shared.
cap	The RSS cap defined for the project. See "Attribute to Limit Physical Memory Usage for Projects" on page 120 or the rcapd(1M) man page for information about how to specify memory caps.
at	The total amount of memory that rcapd attempted to page out since the last rcapstat sample.
avgat	The average amount of memory that rcapd attempted to page out during each sample cycle that occurred since the last rcapstat sample. The rate at which rcapd samples collection RSS can be set with rcapadm. See "rcapd Operation Intervals" on page 124.
pg	The total amount of memory that rcapd successfully paged out since the last rcapstat sample.
avgpg	An estimate of the average amount of memory that rcapd successfully paged out during each sample cycle that occurred since the last rcapstat sample. The rate at which rcapd samples process RSS sizes can be set with rcapadm. See "rcapd Operation Intervals" on page 124.

Commands Used With rcapd

Command Reference	Description
rcapstat(1)	Monitors the resource utilization of capped projects.
rcapadm(1M)	Configures the resource capping daemon, displays the current status of the resource capping daemon if it has been configured, and enables or disables resource capping. Also used to set a temporary memory cap.
rcapd(1M)	The resource capping daemon.

CHAPTER 11

Administering the Resource Capping Daemon (Tasks)

This chapter contains procedures for configuring and using the resource capping daemon `rcapd`.

For an overview of `rcapd`, see Chapter 10, "Physical Memory Control Using the Resource Capping Daemon (Overview)."

Setting the Resident Set Size Cap

Define a physical memory resource resident set size (RSS) cap for a project by adding an `rcap.max-rss` attribute to the `project` database entry.

▼ How to Add an `rcap.max-rss` Attribute for a Project

1. **Be superuser, or have the required rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Add this attribute to the `/etc/project` file:**

   `rcap.max-rss=`*value*

Example 11–1 RSS Project Cap

The following line in the `/etc/project` file sets an RSS cap of 10 gigabytes for a project named `db`.

```
db:100::db,root::rcap.max-rss=10737418240
```

Note that the system might round the specified cap value to a page size.

▼ How to Use the `projmod` Command to Add an `rcap.max-rss` Attribute for a Project

1. **Be superuser, or have the required rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Set an `rcap.max-rss` attribute of 10 gigabytes in the `/etc/project` file, in this case for a project named `db`.**

   ```
   # projmod -a -K rcap.max-rss=10GB db
   ```

   The `/etc/project` file then contains the line:

   ```
   db:100::db,root::rcap.max-rss=10737418240
   ```

Configuring and Using the Resource Capping Daemon (Task Map)

Task	Description	For Instructions
Set the memory cap enforcement threshold.	Configure a cap that will be enforced when the physical memory available to processes is low.	"How to Set the Memory Cap Enforcement Threshold" on page 129
Set the operation interval.	The interval is applied to the periodic operations performed by the resource capping daemon.	"How to Set Operation Intervals" on page 129
Enable resource capping.	Activate resource capping on your system.	"How to Enable Resource Capping" on page 130
Disable resource capping.	Deactivate resource capping on your system.	"How to Disable Resource Capping" on page 130
Report cap and project information.	View example commands for producing reports.	"Reporting Cap and Project Information" on page 132
Monitor a project's resident set size.	Produce a report on the resident set size of a project.	"Monitoring the RSS of a Project" on page 132
Determine a project's working set size.	Produce a report on the working set size of a project.	"Determining the Working Set Size of a Project" on page 133

Task	Description	For Instructions
Report on memory utilization and memory caps.	Print a memory utilization and cap enforcement line at the end of the report for each interval.	"Reporting Memory Utilization and the Memory Cap Enforcement Threshold" on page 134

Administering the Resource Capping Daemon With `rcapadm`

This section contains procedures for configuring the resource capping daemon with `rcapadm`. See "`rcapd` Configuration" on page 121 and the `rcapadm(1M)` man page for more information. Using the `rcapadm` to specify a temporary resource cap for a zone is also covered.

If used without arguments, `rcapadm` displays the current status of the resource capping daemon if it has been configured.

▼ How to Set the Memory Cap Enforcement Threshold

Caps can be configured so that they will not be enforced until the physical memory available to processes is low. See "Memory Cap Enforcement Threshold" on page 122 for more information.

The minimum (and default) value is `0`, which means that memory caps are always enforced. To set a different minimum, follow this procedure.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Use the `-c` option of `rcapadm` to set a different physical memory utilization value for memory cap enforcement.**

```
# rcapadm -c percent
```

percent is in the range 0 to 100. Higher values are less restrictive. A higher value means capped project workloads can execute without having caps enforced until the system's memory utilization exceeds this threshold.

See Also To display the current physical memory utilization and the cap enforcement threshold, see "Reporting Memory Utilization and the Memory Cap Enforcement Threshold" on page 134.

▼ How to Set Operation Intervals

"`rcapd` Operation Intervals" on page 124 contains information about the intervals for the periodic operations performed by `rcapd`. To set operation intervals using `rcapadm`, follow this procedure.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Use the `-i` option to set interval values.**

```
# rcapadm -i interval=value,...,interval=value
```

Note – All interval values are specified in seconds.

How to Enable Resource Capping

There are three ways to enable resource capping on your system. Enabling resource capping also sets the `/etc/rcap.conf` file with default values.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Enable the resource capping daemon in one of the following ways:**

- Turn on resource capping using the `svcadm` command.

  ```
  # svcadm enable rcap
  ```

- Enable the resource capping daemon so that it will be started now and also be started each time the system is booted:

  ```
  # rcapadm -E
  ```

- Enable the resource capping daemon at boot without starting it now by also specifying the `-n` option:

  ```
  # rcapadm -n -E
  ```

How to Disable Resource Capping

There are three ways to disable resource capping on your system.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Disable the resource capping daemon in one of the following ways:**

- **Turn off resource capping using the `svcadm` command.**

  ```
  # svcadm disable rcap
  ```

- **To disable the resource capping daemon so that it will be stopped now and not be started when the system is booted, type:**

  ```
  # rcapadm -D
  ```

- **To disable the resource capping daemon without stopping it, also specify the `-n` option:**

  ```
  # rcapadm -n -D
  ```

Tip – Disabling the Resource Capping Daemon Safely

Use `rcapadm -D` to safely disable `rcapd`. If the daemon is killed (see the `kill(1)` man page), processes might be left in a stopped state and need to be manually restarted. To resume a process running, use the `prun` command. See the `prun(1)` man page for more information.

▼ How to Specify a Temporary Resource Cap for a Zone

This procedure is use to allocate the maximum amount of memory that can be consumed by a specified zone. This value lasts only until the next reboot. To set a persistent cap, use the `zonecfg` command.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Set a maximum memory value of 512 megabytes for the zone `my-zone`.**

```
# rcapadm -z testzone -m 512M
```

Producing Reports With `rcapstat`

Use `rcapstat` to report resource capping statistics. "Monitoring Resource Utilization With `rcapstat`" on page 125 explains how to use the `rcapstat` command to generate reports. That section also describes the column headings in the report. The `rcapstat(1)` man page also contains this information.

The following subsections use examples to illustrate how to produce reports for specific purposes.

Reporting Cap and Project Information

In this example, caps are defined for two projects associated with two users. `user1` has a cap of 50 megabytes, and `user2` has a cap of 10 megabytes.

The following command produces five reports at 5-second sampling intervals.

```
user1machine% rcapstat 5 5
    id project  nproc     vm    rss   cap    at avgat    pg avgpg
112270   user1     24   123M    35M   50M   50M    0K 3312K    0K
 78194   user2      1  2368K  1856K   10M    0K    0K    0K    0K
    id project  nproc     vm    rss   cap    at avgat    pg avgpg
112270   user1     24   123M    35M   50M    0K    0K    0K    0K
 78194   user2      1  2368K  1856K   10M    0K    0K    0K    0K
    id project  nproc     vm    rss   cap    at avgat    pg avgpg
112270   user1     24   123M    35M   50M    0K    0K    0K    0K
 78194   user2      1  2368K  1928K   10M    0K    0K    0K    0K
    id project  nproc     vm    rss   cap    at avgat    pg avgpg
112270   user1     24   123M    35M   50M    0K    0K    0K    0K
 78194   user2      1  2368K  1928K   10M    0K    0K    0K    0K
    id project  nproc     vm    rss   cap    at avgat    pg avgpg
112270   user1     24   123M    35M   50M    0K    0K    0K    0K
 78194   user2      1  2368K  1928K   10M    0K    0K    0K    0K
```

The first three lines of output constitute the first report, which contains the cap and project information for the two projects and paging statistics since `rcapd` was started. The `at` and `pg` columns are a number greater than zero for `user1` and zero for `user2`, which indicates that at some time in the daemon's history, `user1` exceeded its cap but `user2` did not.

The subsequent reports show no significant activity.

Monitoring the RSS of a Project

The following example uses project `user1`, which has an RSS in excess of its RSS cap.

The following command produces five reports at 5-second sampling intervals.

```
user1machine% rcapstat 5 5

    id project  nproc    vm   rss   cap    at avgat     pg  avgpg
376565   user1      3 6249M 6144M 6144M  690M  220M  5528K  2764K
376565   user1      3 6249M 6144M 6144M    0M  131M  4912K  1637K
376565   user1      3 6249M 6171M 6144M   27M  147M  6048K  2016K
376565   user1      3 6249M 6146M 6144M 4872M  174M  4368K  1456K
376565   user1      3 6249M 6156M 6144M   12M  161M  3376K  1125K
```

The `user1` project has three processes that are actively using physical memory. The positive values in the `pg` column indicate that `rcapd` is consistently paging out memory as it attempts to meet the cap by lowering the physical memory utilization of the project's processes. However, `rcapd` does not succeed in keeping the RSS below the cap value. This is indicated by the varying rss values that do not show a corresponding decrease. As soon as memory is paged out, the workload uses it again and the RSS count goes back up. This means that all of the project's resident memory is being actively used and the working set size (*WSS*) is greater than the cap. Thus, rcapd is forced to page out some of the working set to meet the cap. Under this condition, the system will continue to experience high page fault rates, and associated I/O, until one of the following occurs:

- The WSS becomes smaller.
- The cap is raised.
- The application changes its memory access pattern.

In this situation, shortening the sample interval might reduce the discrepancy between the RSS value and the cap value by causing rcapd to sample the workload and enforce caps more frequently.

Note – A page fault occurs when either a new page must be created or the system must copy in a page from a swap device.

Determining the Working Set Size of a Project

The following example is a continuation of the previous example, and it uses the same project.

The previous example shows that the user1 project is using more physical memory than its cap allows. This example shows how much memory the project workload requires.

```
user1machine% rcapstat 5 5
    id project  nproc    vm   rss   cap    at avgat     pg  avgpg
376565   user1      3 6249M 6144M 6144M  690M    0K   689M     0K
376565   user1      3 6249M 6144M 6144M    0K    0K     0K     0K
376565   user1      3 6249M 6171M 6144M   27M    0K    27M     0K
376565   user1      3 6249M 6146M 6144M 4872K    0K  4816K     0K
376565   user1      3 6249M 6156M 6144M   12M    0K    12M     0K
376565   user1      3 6249M 6150M 6144M 5848K    0K  5816K     0K
376565   user1      3 6249M 6155M 6144M   11M    0K    11M     0K
376565   user1      3 6249M 6150M   10G   32K    0K    32K     0K
376565   user1      3 6249M 6214M   10G    0K    0K     0K     0K
376565   user1      3 6249M 6247M   10G    0K    0K     0K     0K
376565   user1      3 6249M 6247M   10G    0K    0K     0K     0K
376565   user1      3 6249M 6247M   10G    0K    0K     0K     0K
376565   user1      3 6249M 6247M   10G    0K    0K     0K     0K
376565   user1      3 6249M 6247M   10G    0K    0K     0K     0K
376565   user1      3 6249M 6247M   10G    0K    0K     0K     0K
```

Halfway through the cycle, the cap on the user1 project was increased from 6 gigabytes to 10 gigabytes. This increase stops cap enforcement and allows the resident set size to grow, limited only by other processes and the amount of memory in the machine. The rss column might stabilize to reflect the project working set size (WSS), 6247M in this example. This is the minimum cap value that allows the project's processes to operate without continuously incurring page faults.

While the cap on `user1` is 6 gigabytes, in every 5–second sample interval the RSS decreases and I/O increases as `rcapd` pages out some of the workload's memory. Shortly after a page out completes, the workload, needing those pages, pages them back in as it continues running. This cycle repeats until the cap is raised to 10 gigabytes, approximately halfway through the example. The RSS then stabilizes at 6.1 gigabytes. Since the workload's RSS is now below the cap, no more paging occurs. The I/O associated with paging stops as well. Thus, the project required 6.1 gigabytes to perform the work it was doing at the time it was being observed.

Also see the `vmstat(1M)` and `iostat(1M)` man pages.

Reporting Memory Utilization and the Memory Cap Enforcement Threshold

You can use the `-g` option of `rcapstat` to report the following:

- Current physical memory utilization as a percentage of physical memory installed on the system
- System memory cap enforcement threshold set by `rcapadm`

The `-g` option causes a memory utilization and cap enforcement line to be printed at the end of the report for each interval.

```
# rcapstat -g
    id project   nproc    vm   rss   cap    at avgat   pg  avgpg
376565    rcap       0    0K    0K   10G    0K    0K   0K     0K
physical memory utilization: 55%   cap enforcement threshold: 0%
    id project   nproc    vm   rss   cap    at avgat   pg  avgpg
376565    rcap       0    0K    0K   10G    0K    0K   0K     0K
physical memory utilization: 55%   cap enforcement threshold: 0%
```

CHAPTER 12

Resource Pools (Overview)

This chapter discusses the following features:

- Resource pools, which are used for partitioning machine resources
- Dynamic resource pools (DRPs), which dynamically adjust each resource pool's resource allocation to meet established system goals

Resource pools and dynamic resource pools are services in the Oracle Solaris service management facility (SMF). Each of these services is enabled separately.

The following topics are covered in this chapter:

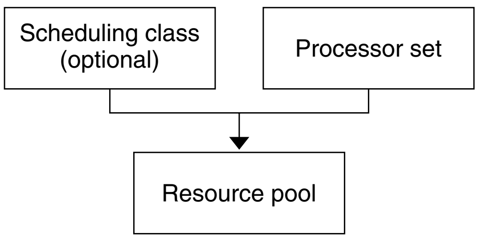
For procedures using this functionality, see Chapter 13, "Creating and Administering Resource Pools (Tasks)."

Introduction to Resource Pools

Resource pools enable you to separate workloads so that workload consumption of certain resources does not overlap. This resource reservation helps to achieve predictable performance on systems with mixed workloads.

Resource pools provide a persistent configuration mechanism for processor set (pset) configuration and, optionally, scheduling class assignment.

FIGURE 12–1 Resource Pool Framework



A pool can be thought of as a specific binding of the various resource sets that are available on your system. You can create pools that represent different kinds of possible resource combinations:

```
pool1: pset_default
pool2: pset1
pool3: pset1, pool.scheduler="FSS"
```

By grouping multiple partitions, pools provide a handle to associate with labeled workloads. Each project entry in the `/etc/project` file can have a single pool associated with that entry, which is specified using the `project.pool` attribute.

When pools are enabled, a *default pool* and a *default processor set* form the base configuration. Additional user-defined pools and processor sets can be created and added to the configuration. A CPU can only belong to one processor set. User-defined pools and processor sets can be destroyed. The default pool and the default processor set cannot be destroyed.

The default pool has the `pool.default` property set to `true`. The default processor set has the `pset.default` property set to `true`. Thus, both the default pool and the default processor set can be identified even if their names have been changed.

The user-defined pools mechanism is primarily for use on large machines of more than four CPUs. However, small machines can still benefit from this functionality. On small machines, you can create pools that share noncritical resource partitions. The pools are separated only on the basis of critical resources.

Introduction to Dynamic Resource Pools

Dynamic resource pools provide a mechanism for dynamically adjusting each pool's resource allocation in response to system events and application load changes. DRPs simplify and reduce the number of decisions required from an administrator. Adjustments are automatically made to preserve the system performance goals specified by an administrator. The changes made to the configuration are logged. These features are primarily enacted through the resource controller `poold`, a system daemon that should always be active when dynamic resource allocation is required. Periodically, `poold` examines the load on the system and determines whether intervention is required to enable the system to maintain optimal performance with respect to resource consumption. The `poold` configuration is held in the `libpool` configuration. For more information on `poold`, see the `poold`(1M) man page.

About Enabling and Disabling Resource Pools and Dynamic Resource Pools

To enable and disable resource pools and dynamic resource pools, see "Enabling and Disabling the Pools Facility" on page 159.

Resource Pools Used in Zones

As an alternative to associating a zone with a configured resource pool on your system, you can use the `zonecfg` command to create a temporary pool that is in effect while the zone is running. See "`dedicated-cpu` Resource" on page 207 for more information.

On a system that has zones enabled, a non-global zone can be associated with one resource pool, although the pool need not be exclusively assigned to a particular zone. Moreover, you cannot bind individual processes in non-global zones to a different pool by using the `poolbind` command from the global zone. To associate a non-global zone with a pool, see "Configuring, Verifying, and Committing a Zone" on page 238.

Note that if you set a scheduling class for a pool and you associate a non-global zone with that pool, the zone uses that scheduling class by default.

If you are using dynamic resource pools, the scope of an executing instance of `poold` is limited to the global zone.

The `poolstat` utility run in a non-global zone displays only information about the pool associated with the zone. The `pooladm` command run without arguments in a non-global zone displays only information about the pool associated with the zone.

For information about resource pool commands, see "Commands Used With the Resource Pools Facility" on page 155.

When to Use Pools

Resource pools offer a versatile mechanism that can be applied to many administrative scenarios.

Batch compute server	Use pools functionality to split a server into two pools. One pool is used for login sessions and interactive work by timesharing users. The other pool is used for jobs that are submitted through the batch system.
Application or database server	Partition the resources for interactive applications in accordance with the applications' requirements.
Turning on applications in phases	Set user expectations. You might initially deploy a machine that is running only a fraction of the services that the machine is ultimately expected to deliver. User difficulties can occur if reservation-based resource management mechanisms are not established when the machine comes online. For example, the fair share scheduler optimizes CPU utilization. The response times for a machine that is running only one application can be misleadingly fast. Users will not see these response times with multiple applications loaded. By using separate pools for each application, you can place a ceiling on the number of CPUs available to each application before you deploy all applications.
Complex timesharing server	Partition a server that supports large user populations. Server partitioning provides an isolation mechanism that leads to a more predictable per-user response. By dividing users into groups that bind to separate pools, and using the fair share scheduling (FSS) facility, you can tune CPU allocations to favor sets of users that have priority. This assignment can be based on user role, accounting chargeback, and so forth.
Workloads that change seasonally	Use resource pools to adjust to changing demand. Your site might experience predictable shifts in workload demand over long periods of time, such as monthly, quarterly, or annual cycles. If your site experiences these shifts, you can alternate between

	multiple pools configurations by invoking `pooladm` from a `cron` job. (See "Resource Pools Framework" on page 139.)
Real-time applications	Create a real-time pool by using the RT scheduler and designated processor resources.
System utilization	Enforce system goals that you establish. Use the automated pools daemon feature to identify available resources and then monitor workloads to detect when your specified objectives are no longer being satisfied. The daemon can take corrective action if possible, or the condition can be logged.

Resource Pools Framework

The `/etc/pooladm.conf` configuration file describes the static pools configuration. A static configuration represents the way in which an administrator would like a system to be configured with respect to resource pools functionality. An alternate file name can be specified.

When the service management facility (SMF) or the `pooladm -e` command is used to enable the resource pools framework, then, if an `/etc/pooladm.conf` file exists, the configuration contained in the file is applied to the system.

The kernel holds information about the disposition of resources within the resource pools framework. This is known as the dynamic configuration, and it represents the resource pools functionality for a particular system at a point in time. The dynamic configuration can be viewed by using the `pooladm` command. Note that the order in which properties are displayed for pools and resource sets can vary. Modifications to the dynamic configuration are made in the following ways:

- Indirectly, by applying a static configuration file
- Directly, by using the `poolcfg` command with the `-d` option

More than one static pools configuration file can exist, for activation at different times. You can alternate between multiple pools configurations by invoking `pooladm` from a `cron` job. See the `cron(1M)` man page for more information on the `cron` utility.

By default, the resource pools framework is not active. Resource pools must be enabled to create or modify the dynamic configuration. Static configuration files can be manipulated with the `poolcfg` or `libpool` commands even if the resource pools framework is disabled. Static configuration files cannot be created if the pools facility is not active. For more information on the configuration file, see "Creating Pools Configurations" on page 142.

The commands used with resource pools and the `poold` system daemon are described in the following man pages:

- `pooladm`(1M)
- `poolbind`(1M)
- `poolcfg`(1M)
- `poold`(1M)
- `poolstat`(1M)
- `libpool`(3LIB)

`/etc/pooladm.conf` Contents

All resource pool configurations, including the dynamic configuration, can contain the following elements.

system	Properties affecting the total behavior of the system
pool	A resource pool definition
pset	A processor set definition
cpu	A processor definition

All of these elements have properties that can be manipulated to alter the state and behavior of the resource pools framework. For example, the pool property `pool.importance` indicates the relative importance of a given pool. This property is used for possible resource dispute resolution. For more information, see `libpool`(3LIB).

Pools Properties

The pools facility supports named, typed properties that can be placed on a pool, resource, or component. Administrators can store additional properties on the various pool elements. A property namespace similar to the project attribute is used.

For example, the following comment indicates that a given pset is associated with a particular `Datatree` database.

```
Datatree,pset.dbname=warehouse
```

For additional information about property types, see “`poold` Properties” on page 147.

Note – A number of special properties are reserved for internal use and cannot be set or removed. See the `libpool`(3LIB) man page for more information.

Implementing Pools on a System

User-defined pools can be implemented on a system by using one of these methods.

- When the Oracle Solaris software boots, an `init` script checks to see if the `/etc/pooladm.conf` file exists. If this file is found and pools are enabled, then `pooladm` is invoked to make this configuration the active pools configuration. The system creates a dynamic configuration to reflect the organization that is requested in `/etc/pooladm.conf`, and the machine's resources are partitioned accordingly.
- When the Oracle Solaris system is running, a pools configuration can either be activated if it is not already present, or modified by using the `pooladm` command. By default, the `pooladm` command operates on `/etc/pooladm.conf`. However, you can optionally specify an alternate location and file name, and use that file to update the pools configuration.

For information about enabling and disabling resource pools, see "Enabling and Disabling the Pools Facility" on page 159. The pools facility cannot be disabled when there are user-defined pools or resources in use.

To configure resource pools, you must have superuser privileges or have the required rights profile.

The `poold` resource controller is started with the dynamic resource pools facility.

`project.pool` Attribute

The `project.pool` attribute can be added to a project entry in the `/etc/project` file to associate a single pool with that entry. New work that is started on a project is bound to the appropriate pool. See Chapter 2, "Projects and Tasks (Overview)," for more information.

For example, you can use the `projmod` command to set the `project.pool` attribute for the project *sales* in the `/etc/project` file:

```
# projmod -a -K project.pool=mypool sales
```

SPARC: Dynamic Reconfiguration Operations and Resource Pools

Dynamic Reconfiguration (DR) enables you to reconfigure hardware while the system is running. A DR operation can increase, reduce, or have no effect on a given type of resource.

Because DR can affect available resource amounts, the pools facility must be included in these operations. When a DR operation is initiated, the pools framework acts to validate the configuration.

If the DR operation can proceed without causing the current pools configuration to become invalid, then the private configuration file is updated. An invalid configuration is one that cannot be supported by the available resources.

If the DR operation would cause the pools configuration to be invalid, then the operation fails and you are notified by a message to the message log. If you want to force the configuration to completion, you must use the DR force option. The pools configuration is then modified to comply with the new resource configuration. For information on the DR process and the force option, see the dynamic reconfiguration user guide for your Sun hardware.

If you are using dynamic resource pools, note that it is possible for a partition to move out of `poold` control while the daemon is active. For more information, see "Identifying a Resource Shortage" on page 152.

Creating Pools Configurations

The configuration file contains a description of the pools to be created on the system. The file describes the elements that can be manipulated.

- system
- pool
- pset
- cpu

See `poolcfg`(1M) for more information on elements that be manipulated.

When pools are enabled, you can create a structured `/etc/pooladm.conf` file in two ways.

- You can use the `pooladm` command with the `-s` option to discover the resources on the current system and place the results in a configuration file.

  This method is preferred. All active resources and components on the system that are capable of being manipulated by the pools facility are recorded. The resources include existing processor set configurations. You can then modify the configuration to rename the processor sets or to create additional pools if necessary.
- You can use the `poolcfg` command with the `-c` option and the `discover` or `create system` *name* subcommands to create a new pools configuration.

  These options are maintained for backward compatibility with previous releases.

Use `poolcfg` or `libpool` to modify the `/etc/pooladm.conf` file. Do not directly edit this file.

Directly Manipulating the Dynamic Configuration

It is possible to directly manipulate CPU resource types in the dynamic configuration by using the poolcfg command with the -d option. There are two methods used to transfer resources.

- You can make a general request to transfer any available identified resources between sets.
- You can transfer resources with specific IDs to a target set. Note that the system IDs associated with resources can change when the resource configuration is altered or after a system reboot.

For an example, see "Transferring Resources" on page 172.

If DRP is in use, note that the resource transfer might trigger action from poold. See "poold Overview" on page 143 for more information.

poold Overview

The pools resource controller, poold, uses system targets and observable statistics to preserve the system performance goals that you specify. This system daemon should always be active when dynamic resource allocation is required.

The poold resource controller identifies available resources and then monitors workloads to determine when the system usage objectives are no longer being met. poold then considers alternative configurations in terms of the objectives, and remedial action is taken. If possible, the resources are reconfigured so that objectives can be met. If this action is not possible, the daemon logs that user-specified objectives can no longer be achieved. Following a reconfiguration, the daemon resumes monitoring workload objectives.

poold maintains a decision history that it can examine. The decision history is used to eliminate reconfigurations that historically did not show improvements.

Note that a reconfiguration can also be triggered asynchronously if the workload objectives are changed or if the resources available to the system are modified.

Managing Dynamic Resource Pools

The DRP service is managed by the service management facility (SMF) under the service identifier svc:/system/pools/dynamic.

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using the svcadm command. The service's status can be queried using the svcs command. See the svcs(1) andsvcadm(1M) man pages for more information.

The SMF interface is the preferred method for controlling DRP, but for backward compatibility, the following methods can also be used.

- If dynamic resource allocation is not required, `poold` can be stopped with the `SIGQUIT` or the `SIGTERM` signal. Either of these signals causes `poold` to terminate gracefully.
- Although `poold` will automatically detect changes in the resource or pools configuration, you can also force a reconfiguration to occur by using the `SIGHUP` signal.

Configuration Constraints and Objectives

When making changes to a configuration, `poold` acts on directions that you provide. You specify these directions as a series of constraints and objectives. `poold` uses your specifications to determine the relative value of different configuration possibilities in relation to the existing configuration. `poold` then changes the resource assignments of the current configuration to generate new candidate configurations.

Configuration Constraints

Constraints affect the range of possible configurations by eliminating some of the potential changes that could be made to a configuration. The following constraints, which are specified in the `libpool` configuration, are available.

- The minimum and maximum CPU allocations
- Pinned components that are not available to be moved from a set
- The importance factor of the pool

See the `libpool(3LIB)` man page and "Pools Properties" on page 140 for more information about pools properties.

See "How to Set Configuration Constraints" on page 168 for usage instructions.

`pset.min` Property and `pset.max` Property Constraints

These two properties place limits on the number of processors that can be allocated to a processor set, both minimum and maximum. See Table 12–1 for more details about these properties.

Within these constraints, a resource partition's resources are available to be allocated to other resource partitions in the same Oracle Solaris instance. Access to the resource is obtained by binding to a pool that is associated with the resource set. Binding is performed at login or manually by an administrator who has the `PRIV_SYS_RES_CONFIG` privilege.

`cpu.pinned` Property Constraint

The `cpu-pinned` property indicates that a particular CPU should not be moved by DRP from the processor set in which it is located. You can set this `libpool` property to maximize cache utilization for a particular application that is executing within a processor set.

See Table 12–1 for more details about this property.

`pool.importance` Property Constraint

The `pool.importance` property describes the relative importance of a pool as defined by the administrator.

Configuration Objectives

Objectives are specified similarly to constraints. The full set of objectives is documented in Table 12–1.

There are two categories of objectives.

Workload dependent — A workload-dependent objective is an objective that will vary according to the nature of the workload running on the system. An example is the `utilization` objective. The utilization figure for a resource set will vary according to the nature of the workload that is active in the set.

Workload independent — A workload-independent objective is an objective that does not vary according to the nature of the workload running on the system. An example is the CPU `locality` objective. The evaluated measure of locality for a resource set does not vary with the nature of the workload that is active in the set.

You can define three types of objectives.

Name	Valid Elements	Operators	Values
`wt-load`	`system`	N/A	N/A
`locality`	`pset`	N/A	`loose` | `tight` | `none`
`utilization`	`pset`	`< > ~`	`0–100%`

Objectives are stored in property strings in the `libpool` configuration. The property names are as follows:

- `system.poold.objectives`

- pset.poold.objectives

Objectives have the following syntax:

- objectives = objective [; objective]*
- objective = [n:] keyword [op] [value]

All objectives take an optional importance prefix. The importance acts as a multiplier for the objective and thus increases the significance of its contribution to the objective function evaluation. The range is from 0 to INT64_MAX (9223372036854775807). If not specified, the default importance value is 1.

Some element types support more than one type of objective. An example is pset. You can specify multiple objective types for these elements. You can also specify multiple utilization objectives on a single pset element.

See "How to Define Configuration Objectives" on page 169 for usage examples.

wt-load Objective

The wt-load objective favors configurations that match resource allocations to resource utilizations. A resource set that uses more resources will be given more resources when this objective is active. wt-load means *weighted load*.

Use this objective when you are satisfied with the constraints you have established using the minimum and maximum properties, and you would like the daemon to manipulate resources freely within those constraints.

The locality Objective

The locality objective influences the impact that locality, as measured by locality group (lgroup) data, has upon the selected configuration. An alternate definition for locality is latency. An lgroup describes CPU and memory resources. The lgroup is used by the Oracle Solaris system to determine the distance between resources, using time as the measurement. For more information on the locality group abstraction, see "Locality Groups Overview" in *Programming Interfaces Guide*.

This objective can take one of the following three values:

tight — If set, configurations that maximize resource locality are favored.

loose — If set, configurations that minimize resource locality are favored.

none — If set, the favorableness of a configuration is not influenced by resource locality. This is the default value for the locality objective.

In general, the locality objective should be set to tight. However, to maximize memory bandwidth or to minimize the impact of DR operations on a resource set, you could set this objective to loose or keep it at the default setting of none.

`utilization` Objective

The `utilization` objective favors configurations that allocate resources to partitions that are not meeting the specified utilization objective.

This objective is specified by using operators and values. The operators are as follows:

- `<` The “less than” operator indicates that the specified value represents a maximum target value.
- `>` The “greater than” operator indicates that the specified value represents a minimum target value.
- `~` The “about” operator indicates that the specified value is a target value about which some fluctuation is acceptable.

A pset can only have one utilization objective set for each type of operator.

- If the ~ operator is set, then the < and > operators cannot be set.
- If the < and > operators are set, then the ~ operator cannot be set. Note that the settings of the < operator and the > operator cannot contradict each other.

You can set both a < and a > operator together to create a range. The values will be validated to make sure that they do not overlap.

Configuration Objectives Example

In the following example, `poold` is to assess these objectives for the pset:

- The `utilization` should be kept between 30 percent and 80 percent.
- The `locality` should be maximized for the processor set.
- The objectives should take the default importance of `1`.

EXAMPLE 12–1 `poold` Objectives Example

```
pset.poold.objectives "utilization > 30; utilization < 80; locality tight"
```

See “How to Define Configuration Objectives” on page 169 for additional usage examples.

`poold` Properties

There are four categories of properties:

- Configuration
- Constraint
- Objective
- Objective Parameter

TABLE 12–1 Defined Property Names

Property Name	Type	Category	Description
system.poold.log-level	string	Configuration	Logging level
system.poold.log-location	string	Configuration	Logging location
system.poold.monitor-interval	uint64	Configuration	Monitoring sample interval
system.poold.history-file	string	Configuration	Decision history location
pset.max	uint64	Constraint	Maximum number of CPUs for this processor set
pset.min	uint64	Constraint	Minimum number of CPUs for this processor set
cpu.pinned	bool	Constraint	CPUs pinned to this processor set
system.poold.objectives	string	Objective	Formatted string following poold's objective expression syntax
pset.poold.objectives	string	Objective	Formatted string following poold's expression syntax
pool.importance	int64	Objective parameter	User-assigned importance

poold Features That Can Be Configured

You can configure these aspects of the daemon's behavior.

- Monitoring interval
- Logging level
- Logging location

These options are specified in the pools configuration. You can also control the logging level from the command line by invoking poold.

poold Monitoring Interval

Use the property name system.poold.monitor-interval to specify a value in milliseconds.

poold Logging Information

Three categories of information are provided through logging. These categories are identified in the logs:

- Configuration
- Monitoring
- Optimization

Use the property name system.poold.log-level to specify the logging parameter. If this property is not specified, the default logging level is NOTICE. The parameter levels are hierarchical. Setting a log level of DEBUG will cause poold to log all defined messages. The INFO level provides a useful balance of information for most administrators.

At the command line, you can use the poold command with the -l option and a parameter to specify the level of logging information generated.

The following parameters are available:

- ALERT
- CRIT
- ERR
- WARNING
- NOTICE
- INFO
- DEBUG

The parameter levels map directly onto their syslog equivalents. See "Logging Location" on page 151 for more information about using syslog.

For more information about how to configure poold logging, see "How to Set the poold Logging Level" on page 171.

Configuration Information Logging

The following types of messages can be generated:

ALERT
Problems accessing the libpool configuration, or some other fundamental, unanticipated failure of the libpool facility. Causes the daemon to exit and requires immediate administrative attention.

CRIT
Problems due to unanticipated failures. Causes the daemon to exit and requires immediate administrative attention.

ERR
Problems with the user-specified parameters that control operation, such as unresolvable, conflicting utilization objectives for a resource set. Requires administrative intervention to correct the objectives. poold attempts to take remedial action by ignoring conflicting objectives, but some errors will cause the daemon to exit.

WARNING — Warnings related to the setting of configuration parameters that, while technically correct, might not be suitable for the given execution environment. An example is marking all CPU resources as pinned, which means that poold cannot move CPU resources between processor sets.

DEBUG — Messages containing the detailed information that is needed when debugging configuration processing. This information is not generally used by administrators.

Monitoring Information Logging

The following types of messages can be generated:

CRIT — Problems due to unanticipated monitoring failures. Causes the daemon to exit and requires immediate administrative attention.

ERR — Problems due to unanticipated monitoring error. Could require administrative intervention to correct.

NOTICE — Messages about resource control region transitions.

INFO — Messages about resource utilization statistics.

DEBUG — Messages containing the detailed information that is needed when debugging monitoring processing. This information is not generally used by administrators.

Optimization Information Logging

The following types of messages can be generated:

WARNING — Messages could be displayed regarding problems making optimal decisions. Examples could include resource sets that are too narrowly constrained by their minimum and maximum values or by the number of pinned components.

Messages could be displayed about problems performing an optimal reallocation due to unforseen limitations. Examples could include removing the last processor from a processor set which contains a bound resource consumer.

NOTICE — Messages about usable configurations or configurations that will not be implemented due to overriding decision histories could be displayed.

INFO — Messages about alternate configurations considered could be displayed.

DEBUG — Messages containing the detailed information that is needed when debugging optimization processing. This information is not generally used by administrators.

Logging Location

The `system.poold.log-location` property is used to specify the location for `poold` logged output. You can specify a location of `SYSLOG` for `poold` output (see `syslog`(3C)).

If this property is not specified, the default location for `poold` logged output is `/var/log/pool/poold`.

When `poold` is invoked from the command line, this property is not used. Log entries are written to `stderr` on the invoking terminal.

Log Management With `logadm`

If `poold` is active, the `logadm.conf` file includes an entry to manage the default file `/var/log/pool/poold`. The entry is:

```
/var/log/pool/poold -N -s 512k
```

See the `logadm`(1M) and the `logadm.conf`(4) man pages.

How Dynamic Resource Allocation Works

This section explains the process and the factors that `poold` uses to dynamically allocate resources.

About Available Resources

Available resources are considered to be all of the resources that are available for use within the scope of the `poold` process. The scope of control is at most a single Oracle Solaris instance.

On a system that has zones enabled, the scope of an executing instance of `poold` is limited to the global zone.

Determining Available Resources

Resource pools encompass all of the system resources that are available for consumption by applications.

For a single executing Oracle Solaris instance, a resource of a single type, such as a CPU, must be allocated to a single partition. There can be one or more partitions for each type of resource. Each partition contains a unique set of resources.

For example, a machine with four CPUs and two processor sets can have the following setup:

pset 0: 0 1

pset 1: 2 3

where 0, 1, 2 and 3 after the colon represent CPU IDs. Note that the two processor sets account for all four CPUs.

The same machine cannot have the following setup:

pset 0: 0 1

pset 1: 1 2 3

It cannot have this setup because CPU 1 can appear in only one pset at a time.

Resources cannot be accessed from any partition other than the partition to which they belong.

To discover the available resources, `poold` interrogates the active pools configuration to find partitions. All resources within all partitions are summed to determine the total amount of available resources for each type of resource that is controlled.

This quantity of resources is the basic figure that `poold` uses in its operations. However, there are constraints upon this figure that limit the flexibility that `poold` has to make allocations. For information about available constraints, see "Configuration Constraints" on page 144.

Identifying a Resource Shortage

The control scope for `poold` is defined as the set of available resources for which `poold` has primary responsibility for effective partitioning and management. However, other mechanisms that are allowed to manipulate resources within this control scope can still affect a configuration. If a partition should move out of control while `poold` is active, `poold` tries to restore control through the judicious manipulation of available resources. If `poold` cannot locate additional resources within its scope, then the daemon logs information about the resource shortage.

Determining Resource Utilization

poold typically spends the greatest amount of time observing the usage of the resources within its scope of control. This monitoring is performed to verify that workload-dependent objectives are being met.

For example, for processor sets, all measurements are made across all of the processors in a set. The resource utilization shows the proportion of time that the resource is in use over the sample interval. Resource utilization is displayed as a percentage from 0 to 100.

Identifying Control Violations

The directives described in "Configuration Constraints and Objectives" on page 144 are used to detect the approaching failure of a system to meet its objectives. These objectives are directly related to workload.

A partition that is not meeting user-configured objectives is a control violation. The two types of control violations are synchronous and asynchronous.

- A synchronous violation of an objective is detected by the daemon in the course of its workload monitoring.
- An asynchronous violation of an objective occurs independently of monitoring action by the daemon.

The following events cause asynchronous objective violations:

- Resources are added to or removed from a control scope.
- The control scope is reconfigured.
- The poold resource controller is restarted.

The contributions of objectives that are not related to workload are assumed to remain constant between evaluations of the objective function. Objectives that are not related to workload are only reassessed when a reevaluation is triggered through one of the asynchronous violations.

Determining Appropriate Remedial Action

When the resource controller determines that a resource consumer is short of resources, the initial response is that increasing the resources will improve performance.

Alternative configurations that meet the objectives specified in the configuration for the scope of control are examined and evaluated.

This process is refined over time as the results of shifting resources are monitored and each resource partition is evaluated for responsiveness. The decision history is consulted to eliminate reconfigurations that did not show improvements in attaining the objective function in the past. Other information, such as process names and quantities, are used to further evaluate the relevance of the historical data.

If the daemon cannot take corrective action, the condition is logged. For more information, see "poold Logging Information" on page 149.

Using poolstat to Monitor the Pools Facility and Resource Utilization

The poolstat utility is used to monitor resource utilization when pools are enabled on your system. This utility iteratively examines all of the active pools on a system and reports statistics based on the selected output mode. The poolstat statistics enable you to determine which resource partitions are heavily utilized. You can analyze these statistics to make decisions about resource reallocation when the system is under pressure for resources.

The poolstat utility includes options that can be used to examine specific pools and report resource set-specific statistics.

If zones are implemented on your system and you use poolstat in a non-global zone, information about the resources associated with the zone's pool is displayed.

For more information about the poolstat utility, see the poolstat(1M) man page. For poolstat task and usage information, see "Using poolstat to Report Statistics for Pool-Related Resources" on page 176.

poolstat Output

In default output format, poolstat outputs a heading line and then displays a line for each pool. A pool line begins with the pool ID and the name of the pool, followed by a column of statistical data for the processor set attached to the pool. Resource sets attached to more than one pool are listed multiple times, once for each pool.

The column headings are as follows:

`id` Pool ID.

`pool` Pool name.

`rid` Resource set ID.

`rset` Resource set name.

`type`	Resource set type.
`min`	Minimum resource set size.
`max`	Maximum resource set size.
`size`	Current resource set size.
`used`	Measure of how much of the resource set is currently used. This usage is calculated as the percentage of utilization of the resource set multiplied by the size of the resource set. If a resource set has been reconfigured during the last sampling interval, this value might be not reported. An unreported value appears as a hyphen (`-`).
`load`	Absolute representation of the load that is put on the resource set. For more information about this property, see the `libpool`(3LIB) man page.

You can specify the following in `poolstat` output:

- The order of the columns
- The headings that appear

Tuning `poolstat` Operation Intervals

You can customize the operations performed by `poolstat`. You can set the sampling interval for the report and specify the number of times that statistics are repeated:

interval — Tune the intervals for the periodic operations performed by `poolstat`. All intervals are specified in seconds.

count — Specify the number of times that the statistics are repeated. By default, `poolstat` reports statistics only once.

If *interval* and *count* are not specified, statistics are reported once. If *interval* is specified and *count* is not specified, then statistics are reported indefinitely.

Commands Used With the Resource Pools Facility

The commands described in the following table provide the primary administrative interface to the pools facility. For information on using these commands on a system that has zones enabled, see "Resource Pools Used in Zones" on page 137.

Man Page Reference	Description
pooladm(1M)	Enables or disables the pools facility on your system. Activates a particular configuration or removes the current configuration and returns associated resources to their default status. If run without options, pooladm prints out the current dynamic pools configuration.
poolbind(1M)	Enables the manual binding of projects, tasks, and processes to a resource pool.
poolcfg(1M)	Provides configuration operations on pools and sets. Configurations created using this tool are instantiated on a target host by using pooladm. If run with the info subcommand argument to the -c option, poolcfg displays information about the static configuration at /etc/pooladm.conf. If a file name argument is added, this command displays information about the static configuration held in the named file. For example, poolcfg -c info /tmp/newconfig displays information about the static configuration contained in the file /tmp/newconfig.
poold(1M)	The pools system daemon. The daemon uses system targets and observable statistics to preserve the system performance goals specified by the administrator. If unable to take corrective action when goals are not being met, poold logs the condition.
poolstat(1M)	Displays statistics for pool-related resources. Simplifies performance analysis and provides information that supports system administrators in resource partitioning and repartitioning tasks. Options are provided for examining specified pools and reporting resource set-specific statistics.

A library API is provided by libpool (see the libpool(3LIB) man page). The library can be used by programs to manipulate pool configurations.

CHAPTER 13

Creating and Administering Resource Pools (Tasks)

This chapter describes how to set up and administer resource pools on your system.

For background information about resource pools, see Chapter 12, "Resource Pools (Overview)."

Administering Resource Pools (Task Map)

Task	Description	For Instructions
Enable or disable resource pools.	Activate or disable resource pools on your system.	"Enabling and Disabling the Pools Facility" on page 159
Enable or disable dynamic resource pools.	Activate or disable dynamic resource pools facilities on your system.	"Enabling and Disabling the Pools Facility" on page 159
Create a static resource pools configuration.	Create a static configuration file that matches the current dynamic configuration. For more information, see "Resource Pools Framework" on page 139.	"How to Create a Static Configuration" on page 163
Modify a resource pools configuration.	Revise a pools configuration on your system, for example, by creating additional pools.	"How to Modify a Configuration" on page 164
Associate a resource pool with a scheduling class.	Associate a pool with a scheduling class so that all processes bound to the pool use the specified scheduler.	"How to Associate a Pool With a Scheduling Class" on page 166

Task	Description	For Instructions
Set configuration constraints and define configuration objectives.	Specify objectives for poold to consider when taking corrective action. For more information on configuration objectives, see "poold Overview" on page 143.	"How to Set Configuration Constraints" on page 168 and "How to Define Configuration Objectives" on page 169
Set the logging level.	Specify the level of logging information that poold generates.	"How to Set the poold Logging Level" on page 171
Use a text file with the poolcfg command.	The poolcfg command can take input from a text file.	"How to Use Command Files With poolcfg" on page 171
Transfer resources in the kernel.	Transfer resources in the kernel. For example, transfer resources with specific IDs to a target set.	"Transferring Resources" on page 172
Activate a pools configuration.	Activate the configuration in the default configuration file.	"How to Activate a Pools Configuration" on page 173
Validate a pools configuration before you commit the configuration.	Validate a pools configuration to test what will happen when the validation occurs.	"How to Validate a Configuration Before Committing the Configuration" on page 173
Remove a pools configuration from your system.	All associated resources, such as processor sets, are returned to their default status.	"How to Remove a Pools Configuration" on page 173
Bind processes to a pool.	Manually associate a running process on your system with a resource pool.	"How to Bind Processes to a Pool" on page 174
Bind tasks or projects to a pool.	Associate tasks or projects with a resource pool.	"How to Bind Tasks or Projects to a Pool" on page 175
Bind new processes to a resource pool.	To automatically bind new processes in a project to a given pool, add an attribute to each entry in the project database.	"How to Set the project.pool Attribute for a Project" on page 175
Use project attributes to bind a process to a different pool.	Modify the pool binding for new processes that are started.	"How to Use project Attributes to Bind a Process to a Different Pool" on page 175
Use the poolstat utility to produce reports.	Produce multiple reports at specifed intervals.	"Producing Multiple Reports at Specific Intervals" on page 176
Report resource set statistics.	Use the poolstat utility to report statistics for a pset resource set.	"Reporting Resource Set Statistics" on page 177

Enabling and Disabling the Pools Facility

You can enable and disable the resource pools and dynamic resource pools services on your system by using the `svcadm` command described in the `svcadm`(1M) man page.

You can also use the `pooladm` command described in the `pooladm`(1M) man page to perform the following tasks:

- Enable the pools facility so that pools can be manipulated
- Disable the pools facility so that pools cannot be manipulated

---

Note – When a system is upgraded, if the resource pools framework is enabled and an `/etc/pooladm.conf` file exists, the pools service is enabled and the configuration contained in the file is applied to the system.

▼ How to Enable the Resource Pools Service Using `svcadm`

1. **Be superuser, or have the required rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Enable the resource pools service.**

   ```
   # svcadm enable system/pools:default
   ```

▼ How to Disable the Resource Pools Service Using `svcadm`

1. **Be superuser, or have the required rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Disable the resource pools service.**

   ```
   # svcadm disable system/pools:default
   ```

▼ How to Enable the Dynamic Resource Pools Service Using `svcadm`

1. **Be superuser, or have the required rights profile.**

2. **Enable the dynamic resource pools service.**

```
# svcadm enable system/pools/dynamic:default
```

Example 13–1 Dependency of the Dynamic Resource Pools Service on the Resource Pools Service

This example shows that you must first enable resource pools if you want to run DRP.

There is a dependency between resource pools and dynamic resource pools. DRP is now a dependent service of resource pools. DRP can be independently enabled and disabled apart from resource pools.

The following display shows that both resource pools and dynamic resource pools are currently disabled:

```
# svcs *pool*
STATE          STIME    FMRI
disabled       10:32:26 svc:/system/pools/dynamic:default
disabled       10:32:26 svc:/system/pools:default
```

Enable dynamic resource pools :

```
# svcadm enable svc:/system/pools/dynamic:default
# svcs -a | grep pool
disabled       10:39:00 svc:/system/pools:default
offline        10:39:12 svc:/system/pools/dynamic:default
```

Note that the DRP service is still offline.

Use the `-x` option of the `svcs` command to determine why the DRP service is offline:

```
# svcs -x *pool*
svc:/system/pools:default (resource pools framework)
 State: disabled since Wed 25 Jan 2006 10:39:00 AM GMT
Reason: Disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-05
   See: libpool(3LIB)
   See: pooladm(1M)
   See: poolbind(1M)
   See: poolcfg(1M)
   See: poolstat(1M)
   See: /var/svc/log/system-pools:default.log
Impact: 1 dependent service is not running.  (Use -v for list.)

svc:/system/pools/dynamic:default (dynamic resource pools)
 State: offline since Wed 25 Jan 2006 10:39:12 AM GMT
```

```
Reason: Service svc:/system/pools:default is disabled.
   See: http://sun.com/msg/SMF-8000-GE
   See: poold(1M)
   See: /var/svc/log/system-pools-dynamic:default.log
Impact: This service is not running.
```

Enable the resource pools service so that the DRP service can run:

```
# svcadm enable svc:/system/pools:default
```

When the `svcs *pool*` command is used, the system displays:

```
# svcs *pool*
STATE          STIME    FMRI
online         10:40:27 svc:/system/pools:default
online         10:40:27 svc:/system/pools/dynamic:default
```

Example 13–2 Effect on Dynamic Resource Pools When the Resource Pools Service Is Disabled

If both services are online and you disable the resource pools service:

```
# svcadm disable svc:/system/pools:default
```

When the `svcs *pool*` command is used, the system displays:

```
# svcs *pool*
STATE          STIME    FMRI
disabled       10:41:05 svc:/system/pools:default
online         10:40:27 svc:/system/pools/dynamic:default
# svcs *pool*
STATE          STIME    FMRI
disabled       10:41:05 svc:/system/pools:default
online         10:40:27 svc:/system/pools/dynamic:default
```

But eventually, the DRP service moves to `offline` because the resource pools service has been disabled:

```
# svcs *pool*
STATE          STIME    FMRI
disabled       10:41:05 svc:/system/pools:default
offline        10:41:12 svc:/system/pools/dynamic:default
```

Determine why the DRP service is offline:

```
# svcs -x *pool*
svc:/system/pools:default (resource pools framework)
 State: disabled since Wed 25 Jan 2006 10:41:05 AM GMT
Reason: Disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-05
   See: libpool(3LIB)
   See: pooladm(1M)
   See: poolbind(1M)
   See: poolcfg(1M)
```

```
   See: poolstat(1M)
   See: /var/svc/log/system-pools:default.log
Impact: 1 dependent service is not running.  (Use -v for list.)

svc:/system/pools/dynamic:default (dynamic resource pools)
 State: offline since Wed 25 Jan 2006 10:41:12 AM GMT
Reason: Service svc:/system/pools:default is disabled.
   See: http://sun.com/msg/SMF-8000-GE
   See: poold(1M)
   See: /var/svc/log/system-pools-dynamic:default.log
Impact: This service is not running.
```

Resource pools must be started for DRP to work. For example, resource pools could be started by using the `pooladm` command with the `-e` option:

```
# pooladm -e
```

Then the `svcs *pool*` command displays:

```
# svcs *pool*
STATE          STIME    FMRI
online         10:42:23 svc:/system/pools:default
online         10:42:24 svc:/system/pools/dynamic:default
```

▼ How to Disable the Dynamic Resource Pools Service Using `svcadm`

1. **Be superuser, or have the required rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Disable the dynamic resource pools service.**

   ```
   # svcadm disable system/pools/dynamic:default
   ```

▼ How to Enable Resource Pools Using `pooladm`

1. **Be superuser, or have the required rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Enable the pools facility.**

   ```
   # pooladm -e
   ```

How to Disable Resource Pools Using pooladm

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Disable the pools facility.**

```
# pooladm -d
```

Configuring Pools

How to Create a Static Configuration

Use the `-s` option to `/usr/sbin/pooladm` to create a static configuration file that matches the current dynamic configuration. Unless a different file name is specified, the default location `/etc/pooladm.conf` is used.

Commit your configuration using the `pooladm` command with the `-c` option. Then, use the `pooladm` command with the `-s` option to update the static configuration to match the state of the dynamic configuration.

Note – The new functionality `pooladm -s` is preferred over the previous functionality `poolcfg -c discover` for creating a new configuration that matches the dynamic configuration.

Before You Begin Enable pools on your system.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Update the static configuration file to match the current dynamic configuration.**

```
# pooladm -s
```

3 **View the contents of the configuration file in readable form.**

Note that the configuration contains default elements created by the system.

```
# poolcfg -c info
system tester
        string  system.comment
        int     system.version 1
        boolean system.bind-default true
```

```
        int     system.poold.pid 177916

        pool pool_default
                int     pool.sys_id 0
                boolean pool.active true
                boolean pool.default true
                int     pool.importance 1
                string  pool.comment
                pset    pset_default

        pset pset_default
                int     pset.sys_id -1
                boolean pset.default true
                uint    pset.min 1
                uint    pset.max 65536
                string  pset.units population
                uint    pset.load 10
                uint    pset.size 4
                string  pset.comment
                boolean testnullchanged true

                cpu
                        int     cpu.sys_id 3
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 2
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 1
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 0
                        string  cpu.comment
                        string  cpu.status on-line
```

4 **Commit the configuration at `/etc/pooladm.conf`.**

```
# pooladm -c
```

5 **(Optional) To copy the dynamic configuration to a static configuration file called `/tmp/backup`, type the following:**

```
# pooladm -s /tmp/backup
```

▼ How to Modify a Configuration

To enhance your configuration, create a processor set named `pset_batch` and a pool named `pool_batch`. Then join the pool and the processor set with an association.

Note that you must quote subcommand arguments that contain white space.

1 Be superuser, or have the required rights profile.

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Create processor set `pset_batch`.

```
# poolcfg -c 'create pset pset_batch (uint pset.min = 2; uint pset.max = 10)'
```

3 Create pool `pool_batch`.

```
# poolcfg -c 'create pool pool_batch'
```

4 Join the pool and the processor set with an association.

```
# poolcfg -c 'associate pool pool_batch (pset pset_batch)'
```

5 Display the edited configuration.

```
# poolcfg -c info
system tester
        string  system.comment kernel state
        int     system.version 1
        boolean system.bind-default true
        int     system.poold.pid 177916

        pool pool_default
                int     pool.sys_id 0
                boolean pool.active true
                boolean pool.default true
                int     pool.importance 1
                string  pool.comment
                pset    pset_default

        pset pset_default
                int     pset.sys_id -1
                boolean pset.default true
                uint    pset.min 1
                uint    pset.max 65536
                string  pset.units population
                uint    pset.load 10
                uint    pset.size 4
                string  pset.comment
                boolean testnullchanged true

                cpu
                        int     cpu.sys_id 3
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 2
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 1
                        string  cpu.comment
```

```
                string  cpu.status on-line

        cpu
                int     cpu.sys_id 0
                string  cpu.comment
                string  cpu.status on-line

pool pool_batch
        boolean pool.default false
        boolean pool.active true
        int pool.importance 1
        string pool.comment
        pset pset_batch

pset pset_batch
        int pset.sys_id -2
        string pset.units population
        boolean pset.default true
        uint pset.max 10
        uint pset.min 2
        string pset.comment
        boolean pset.escapable false
        uint pset.load 0
        uint pset.size 0

        cpu
                int     cpu.sys_id 5
                string  cpu.comment
                string  cpu.status on-line

        cpu
                int     cpu.sys_id 4
                string  cpu.comment
                string  cpu.status on-line
```

6 **Commit the configuration at `/etc/pooladm.conf`.**

```
# pooladm -c
```

7 **(Optional) To copy the dynamic configuration to a static configuration file named `/tmp/backup`, type the following:**

```
# pooladm -s /tmp/backup
```

▼ How to Associate a Pool With a Scheduling Class

You can associate a pool with a scheduling class so that all processes bound to the pool use this scheduler. To do this, set the `pool.scheduler` property to the name of the scheduler. This example associates the pool `pool_batch` with the fair share scheduler (FSS).

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Modify pool `pool_batch` to be associated with the FSS.**

```
# poolcfg -c 'modify pool pool_batch (string pool.scheduler="FSS")'
```

3 **Display the edited configuration.**

```
# poolcfg -c info
system tester
        string  system.comment
        int     system.version 1
        boolean system.bind-default true
        int     system.poold.pid 177916

        pool pool_default
                int     pool.sys_id 0
                boolean pool.active true
                boolean pool.default true
                int     pool.importance 1
                string  pool.comment
                pset    pset_default

        pset pset_default
                int     pset.sys_id -1
                boolean pset.default true
                uint    pset.min 1
                uint    pset.max 65536
                string  pset.units population
                uint    pset.load 10
                uint    pset.size 4
                string  pset.comment
                boolean testnullchanged true

                cpu
                        int     cpu.sys_id 3
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 2
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 1
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 0
                        string  cpu.comment
                        string  cpu.status on-line

        pool pool_batch
                boolean pool.default false
                boolean pool.active true
                int pool.importance 1
                string pool.comment
                string pool.scheduler FSS
                pset batch
```

```
pset pset_batch
        int pset.sys_id -2
        string pset.units population
        boolean pset.default true
        uint pset.max 10
        uint pset.min 2
        string pset.comment
        boolean pset.escapable false
        uint pset.load 0
        uint pset.size 0

        cpu
                int     cpu.sys_id 5
                string  cpu.comment
                string  cpu.status on-line

        cpu
                int     cpu.sys_id 4
                string  cpu.comment
                string  cpu.status on-line
```

4 **Commit the configuration at `/etc/pooladm.conf`:**

```
# pooladm -c
```

5 **(Optional) To copy the dynamic configuration to a static configuration file called `/tmp/backup`, type the following:**

```
# pooladm -s /tmp/backup
```

▼ How to Set Configuration Constraints

Constraints affect the range of possible configurations by eliminating some of the potential changes that could be made to a configuration. This procedure shows how to set the `cpu.pinned` property.

In the following examples, `cpuid` is an integer.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Modify the `cpu.pinned` property in the static or dynamic configuration:**

- **Modify the boot-time (static) configuration:**

```
# poolcfg -c 'modify cpu <cpuid> (boolean cpu.pinned = true)'
```

- **Modify the running (dynamic) configuration without modifying the boot-time configuration:**

```
# poolcfg -dc 'modify cpu <cpuid> (boolean cpu.pinned = true)'
```

How to Define Configuration Objectives

You can specify objectives for poold to consider when taking corrective action.

In the following procedure, the wt-load objective is being set so that poold tries to match resource allocation to resource utilization. The locality objective is disabled to assist in achieving this configuration goal.

1. **Be superuser, or have the required rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Modify system tester to favor the wt-load objective.**

```
# poolcfg -c 'modify system tester (string system.poold.objectives="wt-load")'
```

3. **Disable the locality objective for the default processor set.**

```
# poolcfg -c 'modify pset pset_default (string pset.poold.objectives="locality none")'     one line
```

4. **Disable the locality objective for the pset_batch processor set.**

```
# poolcfg -c 'modify pset pset_batch (string pset.poold.objectives="locality none")'     one line
```

5. **Display the edited configuration.**

```
# poolcfg -c info
system tester
        string  system.comment
        int     system.version 1
        boolean system.bind-default true
        int     system.poold.pid 177916
        string  system.poold.objectives wt-load

        pool pool_default
                int     pool.sys_id 0
                boolean pool.active true
                boolean pool.default true
                int     pool.importance 1
                string  pool.comment
                pset    pset_default

        pset pset_default
                int     pset.sys_id -1
                boolean pset.default true
                uint    pset.min 1
                uint    pset.max 65536
                string  pset.units population
                uint    pset.load 10
                uint    pset.size 4
                string  pset.comment
                boolean testnullchanged true
                string  pset.poold.objectives locality none

                cpu
```

```
                int     cpu.sys_id 3
                string  cpu.comment
                string  cpu.status on-line

        cpu
                int     cpu.sys_id 2
                string  cpu.comment
                string  cpu.status on-line

        cpu
                int     cpu.sys_id 1
                string  cpu.comment
                string  cpu.status on-line

        cpu
                int     cpu.sys_id 0
                string  cpu.comment
                string  cpu.status on-line

pool pool_batch
        boolean pool.default false
        boolean pool.active true
        int pool.importance 1
        string pool.comment
        string pool.scheduler FSS
        pset batch

pset pset_batch
        int pset.sys_id -2
        string pset.units population
        boolean pset.default true
        uint pset.max 10
        uint pset.min 2
        string pset.comment
        boolean pset.escapable false
        uint pset.load 0
        uint pset.size 0
        string  pset.poold.objectives locality none

        cpu
                int     cpu.sys_id 5
                string  cpu.comment
                string  cpu.status on-line

        cpu
                int     cpu.sys_id 4
                string  cpu.comment
                string  cpu.status on-line
```

6 Commit the configuration at `/etc/pooladm.conf`.

```
# pooladm -c
```

7 (Optional) To copy the dynamic configuration to a static configuration file called `/tmp/backup`, type the following:

```
# pooladm -s /tmp/backup
```

How to Set the `poold` Logging Level

To specify the level of logging information that `poold` generates, set the `system.poold.log-level` property in the `poold` configuration. The `poold` configuration is held in the `libpool` configuration. For information, see "`poold` Logging Information" on page 149 and the `poolcfg(1M)` and `libpool(3LIB)` man pages.

You can also use the `poold` command at the command line to specify the level of logging information that `poold` generates.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Set the logging level by using the `poold` command with the `-l` option and a parameter, for example, `INFO`.**

```
# /usr/lib/pool/poold -l INFO
```

For information about available parameters, see "`poold` Logging Information" on page 149. The default logging level is `NOTICE`.

How to Use Command Files With `poolcfg`

The `poolcfg` command with the `-f` option can take input from a text file that contains `poolcfg` subcommand arguments to the `-c` option. This method is appropriate when you want a set of operations to be performed. When processing multiple commands, the configuration is only updated if all of the commands succeed. For large or complex configurations, this technique can be more useful than per-subcommand invocations.

Note that in command files, the # character acts as a comment mark for the rest of the line.

1 **Create the input file `poolcmds.txt`.**

```
$ cat > poolcmds.txt
create system tester
create pset pset_batch (uint pset.min = 2; uint pset.max = 10)
create pool pool_batch
associate pool pool_batch (pset pset_batch)
```

2 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

3 **Execute the command:**

```
# /usr/sbin/poolcfg -f poolcmds.txt
```

Transferring Resources

Use the `transfer` subcommand argument to the `-c` option of `poolcfg` with the `-d` option to transfer resources in the kernel. The `-d` option specifies that the command operate directly on the kernel and not take input from a file.

The following procedure moves two CPUs from processor set `pset1` to processor set `pset2` in the kernel.

▼ How to Move CPUs Between Processor Sets

1 Be superuser, or have the required rights profile.

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Move two CPUs from `pset1` to `pset2`.

The `from` and `to` subclauses can be used in any order. Only one `to` and `from` subclause is supported per command.

```
# poolcfg -dc 'transfer 2 from pset pset1 to pset2'
```

Example 13–3 Alternative Method to Move CPUs Between Procesor Sets

If specific known IDs of a resource type are to be transferred, an alternative syntax is provided. For example, the following command assigns two CPUs with IDs `0` and `2` to the `pset_large` processor set:

```
# poolcfg -dc 'transfer to pset pset_large (cpu 0; cpu 2)'
```

More Information Troubleshooting

If a transfer fails because there are not enough resources to match the request or because the specified IDs cannot be located, the system displays an error message.

Activating and Removing Pool Configurations

Use the `pooladm` command to make a particular pool configuration active or to remove the currently active pool configuration. See the `pooladm(1M)` man page for more information about this command.

▼ How to Activate a Pools Configuration

To activate the configuration in the default configuration file, `/etc/pooladm.conf`, invoke `pooladm` with the `-c` option, "commit configuration."

1 **Be superuser or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Commit the configuration at `/etc/pooladm.conf`.**

```
# pooladm -c
```

3 **(Optional) Copy the dynamic configuration to a static configuration file, for example, `/tmp/backup`.**

```
# pooladm -s /tmp/backup
```

▼ How to Validate a Configuration Before Committing the Configuration

You can use the `-n` option with the `-c` option to test what will happen when the validation occurs. The configuration will not actually be committed.

The following command attempts to validate the configuration contained at `/home/admin/newconfig`. Any error conditions encountered are displayed, but the configuration itself is not modified.

1 **Be superuser or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Test the validity of the configuration before committing it.**

```
# pooladm -n -c /home/admin/newconfig
```

▼ How to Remove a Pools Configuration

To remove the current active configuration and return all associated resources, such as processor sets, to their default status, use the `-x` option for "remove configuration."

1 **Be superuser or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Remove the current active configuration.

```
# pooladm -x
```

The `-x` option to `pooladm` removes all user-defined elements from the dynamic configuration. All resources revert to their default states, and all pool bindings are replaced with a binding to the default pool.

More Information

Mixing Scheduling Classes Within a Processor Set

You can safely mix processes in the TS and IA classes in the same processor set. Mixing other scheduling classes within one processor set can lead to unpredictable results. If the use of `pooladm -x` results in mixed scheduling classes within one processor set, use the `priocntl` command to move running processes into a different scheduling class. See "How to Manually Move Processes From the TS Class Into the FSS Class" on page 115. Also see the `priocntl(1)` man page.

Setting Pool Attributes and Binding to a Pool

You can set a `project.pool` attribute to associate a resource pool with a project.

You can bind a running process to a pool in two ways:

- You can use the `poolbind` command described in `poolbind(1M)` command to bind a specific process to a named resource pool.
- You can use the `project.pool` attribute in the `project` database to identify the pool binding for a new login session or a task that is launched through the `newtask` command. See the `newtask(1)`, `projmod(1M)`, and `project(4)` man pages.

▼ How to Bind Processes to a Pool

The following procedure uses `poolbind` with the `-p` option to manually bind a process (in this case, the current shell) to a pool named `ohare`.

1 Be superuser or have the required rights profile.

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Manually bind a process to a pool:

```
# poolbind -p ohare $$
```

3 Verify the pool binding for the process by using `poolbind` with the `-q` option.

```
$ poolbind -q $$
155509 ohare
```

The system displays the process ID and the pool binding.

How to Bind Tasks or Projects to a Pool

To bind tasks or projects to a pool, use the `poolbind` command with the `-i` option. The following example binds all processes in the `airmiles` project to the `laguardia` pool.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Bind all processes in the `airmiles` project to the `laguardia` pool.**

```
# poolbind -i project -p laguardia airmiles
```

How to Set the `project.pool` Attribute for a Project

You can set the `project.pool` attribute to bind a project's processes to a resource pool.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Add a `project.pool` attribute to each entry in the `project` database.**

```
# projmod -a -K project.pool=poolname project
```

How to Use `project` Attributes to Bind a Process to a Different Pool

Assume you have a configuration with two pools that are named `studio` and `backstage`. The `/etc/project` file has the following contents:

```
user.paul:1024::::project.pool=studio
user.george:1024::::project.pool=studio
user.ringo:1024::::project.pool=backstage
passes:1027::paul::project.pool=backstage
```

With this configuration, processes that are started by user `paul` are bound by default to the `studio` pool.

User `paul` can modify the pool binding for processes he starts. `paul` can use `newtask` to bind work to the `backstage` pool as well, by launching in the `passes` project.

1 **Launch a process in the passes project.**

```
$ newtask -l -p passes
```

2 **Use the poolbind command with the -q option to verify the pool binding for the process. Also use a double dollar sign ($$) to pass the process number of the parent shell to the command.**

```
$ poolbind -q $$
6384  pool backstage
```

The system displays the process ID and the pool binding.

Using poolstat to Report Statistics for Pool-Related Resources

The poolstat command is used to display statistics for pool-related resources. See "Using poolstat to Monitor the Pools Facility and Resource Utilization" on page 154 and the poolstat(1M) man page for more information.

The following subsections use examples to illustrate how to produce reports for specific purposes.

Displaying Default poolstat Output

Typing poolstat without arguments outputs a header line and a line of information for each pool. The information line shows the pool ID, the name of the pool, and resource statistics for the processor set attached to the pool.

```
machine% poolstat
                                 pset
       id pool            size used load
        0 pool_default       4  3.6  6.2
        1 pool_sales         4  3.3  8.4
```

Producing Multiple Reports at Specific Intervals

The following command produces three reports at 5-second sampling intervals.

```
machine% poolstat 5 3
                                  pset
 id pool                    size used load
 46 pool_sales                 2  1.2  8.3
  0 pool_default               2  0.4  5.2
                                 pset
 id pool                    size used load
 46 pool_sales                 2  1.4  8.4
```

```
  0 pool_default             2  1.9  2.0
                               pset
 id pool                  size used load
 46 pool_sales               2  1.1  8.0
  0 pool_default             2  0.3  5.0
```

Reporting Resource Set Statistics

The following example uses the `poolstat` command with the `-r` option to report statistics for the processor set resource set. Note that the resource set `pset_default` is attached to more than one pool, so this processor set is listed once for each pool membership.

```
machine% poolstat -r pset
      id pool          type rid rset          min  max size used load
       0 pool_default  pset  -1 pset_default    1  65K    2  1.2  8.3
       6 pool_sales    pset   1 pset_sales      1  65K    2  1.2  8.3
       2 pool_other    pset  -1 pset_default    1  10K    2  0.4  5.2
```

CHAPTER 14

Resource Management Configuration Example

This chapter reviews the resource management framework and describes a hypothetical server consolidation project.

The following topics are covered in this chapter:

Configuration to Be Consolidated

In this example, five applications are being consolidated onto a single system. The target applications have resource requirements that vary, different user populations, and different architectures. Currently, each application exists on a dedicated server that is designed to meet the requirements of the application. The applications and their characteristics are identified in the following table.

Application Description	Characteristics
Application server	Exhibits negative scalability beyond 2 CPUs
Database instance for application server	Heavy transaction processing
Application server in test and development environment	GUI-based, with untested code execution
Transaction processing server	Primary concern is response time
Standalone database instance	Processes a large number of transactions and serves multiple time zones

Consolidation Configuration

The following configuration is used to consolidate the applications onto a single system that has the resource pools and the dynamic resource pools facilities enabled.

- The application server has a two–CPU processor set.
- The database instance for the application server and the standalone database instance are consolidated onto a single processor set of at least four CPUs. The standalone database instance is guaranteed 75 percent of that resource.
- The test and development application server requires the IA scheduling class to ensure UI responsiveness. Memory limitations are imposed to lessen the effects of bad code builds.
- The transaction processing server is assigned a dedicated processor set of at least two CPUs, to minimize response latency.

This configuration covers known applications that are executing and consuming processor cycles in each resource set. Thus, constraints can be established that allow the processor resource to be transferred to sets where the resource is required.

- The `wt-load` objective is set to allow resource sets that are highly utilized to receive greater resource allocations than sets that have low utilization.
- The `locality` objective is set to `tight`, which is used to maximize processor locality.

An additional constraint to prevent utilization from exceeding 80 percent of any resource set is also applied. This constraint ensures that applications get access to the resources they require. Moreover, for the transaction processor set, the objective of maintaining utilization below 80 percent is twice as important as any other objectives that are specified. This importance will be defined in the configuration.

Creating the Configuration

Edit the `/etc/project` database file. Add entries to implement the required resource controls and to map users to resource pools, then view the file.

```
# cat /etc/project
.
.
.
user.app_server:2001:Production Application Server:::project.pool=appserver_pool
user.app_db:2002:App Server DB:::project.pool=db_pool;project.cpu-shares=(privileged,1,deny)
development:2003:Test and development::staff:project.pool=dev_pool;
process.max-address-space=(privileged,536870912,deny)      keep with previous line
user.tp_engine:2004:Transaction Engine:::project.pool=tp_pool
user.geo_db:2005:EDI DB:::project.pool=db_pool;project.cpu-shares=(privileged,3,deny)
.
.
.
```

Note – The development team has to execute tasks in the development project because access for this project is based on a user's group ID (GID).

Create an input file named `pool.host`, which will be used to configure the required resource pools. View the file.

```
# cat pool.host
create system host
create pset dev_pset (uint pset.min = 0; uint pset.max = 2)
create pset tp_pset (uint pset.min = 2; uint pset.max=8)
create pset db_pset (uint pset.min = 4; uint pset.max = 6)
create pset app_pset (uint pset.min = 1; uint pset.max = 2)
create pool dev_pool (string pool.scheduler="IA")
create pool appserver_pool (string pool.scheduler="TS")
create pool db_pool (string pool.scheduler="FSS")
create pool tp_pool (string pool.scheduler="TS")
associate pool dev_pool (pset dev_pset)
associate pool appserver_pool (pset app_pset)
associate pool db_pool (pset db_pset)
associate pool tp_pool (pset tp_pset)
modify system tester (string system.poold.objectives="wt-load")
modify pset dev_pset (string pset.poold.objectives="locality tight; utilization < 80")
modify pset tp_pset (string pset.poold.objectives="locality tight; 2: utilization < 80")
modify pset db_pset (string pset.poold.objectives="locality tight;utilization < 80")
modify pset app_pset (string pset.poold.objectives="locality tight; utilization < 80")
```

Update the configuration using the `pool.host` input file.

```
# poolcfg -f pool.host
```

Make the configuration active.

```
# pooladm -c
```

The framework is now functional on the system.

Enable DRP.

```
# svcadm enable pools/dynamic:default
```

Viewing the Configuration

To view the framework configuration, which also contains default elements created by the system, type:

```
# pooladm
system host
        string  system.comment
```

```
int     system.version 1
boolean system.bind-default true
int     system.poold.pid 177916
string  system.poold.objectives wt-load

pool dev_pool
        int     pool.sys_id 125
        boolean pool.default false
        boolean pool.active true
        int     pool.importance 1
        string  pool.comment
        string  pool.scheduler IA
        pset    dev_pset

pool appserver_pool
        int     pool.sys_id 124
        boolean pool.default false
        boolean pool.active true
        int     pool.importance 1
        string  pool.comment
        string  pool.scheduler TS
        pset    app_pset

pool db_pool
        int     pool.sys_id 123
        boolean pool.default false
        boolean pool.active true
        int     pool.importance 1
        string  pool.comment
        string  pool.scheduler FSS
        pset    db_pset

pool tp_pool
        int     pool.sys_id 122
        boolean pool.default false
        boolean pool.active true
        int     pool.importance 1
        string  pool.comment
        string  pool.scheduler TS
        pset    tp_pset

pool pool_default
        int     pool.sys_id 0
        boolean pool.default true
        boolean pool.active true
        int     pool.importance 1
        string  pool.comment
        string  pool.scheduler TS
        pset    pset_default

pset dev_pset
        int     pset.sys_id 4
        string  pset.units population
        boolean pset.default false
        uint    pset.min 0
        uint    pset.max 2
        string  pset.comment
        boolean pset.escapable false
        uint    pset.load 0
```

```
                uint    pset.size 0
                string  pset.poold.objectives locality tight; utilization < 80

        pset tp_pset
                int     pset.sys_id 3
                string  pset.units population
                boolean pset.default false
                uint    pset.min 2
                uint    pset.max 8
                string  pset.comment
                boolean pset.escapable false
                uint    pset.load 0
                uint    pset.size 0
                string  pset.poold.objectives locality tight; 2: utilization < 80

                cpu
                        int     cpu.sys_id 1
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 2
                        string  cpu.comment
                        string  cpu.status on-line

        pset db_pset
                int     pset.sys_id 2
                string  pset.units population
                boolean pset.default false
                uint    pset.min 4
                uint    pset.max 6
                string  pset.comment
                boolean pset.escapable false
                uint    pset.load 0
                uint    pset.size 0
                string  pset.poold.objectives locality tight; utilization < 80

                cpu
                        int     cpu.sys_id 3
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 4
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 5
                        string  cpu.comment
                        string  cpu.status on-line

                cpu
                        int     cpu.sys_id 6
                        string  cpu.comment
                        string  cpu.status on-line
        pset app_pset
                int     pset.sys_id 1
                string  pset.units population
```

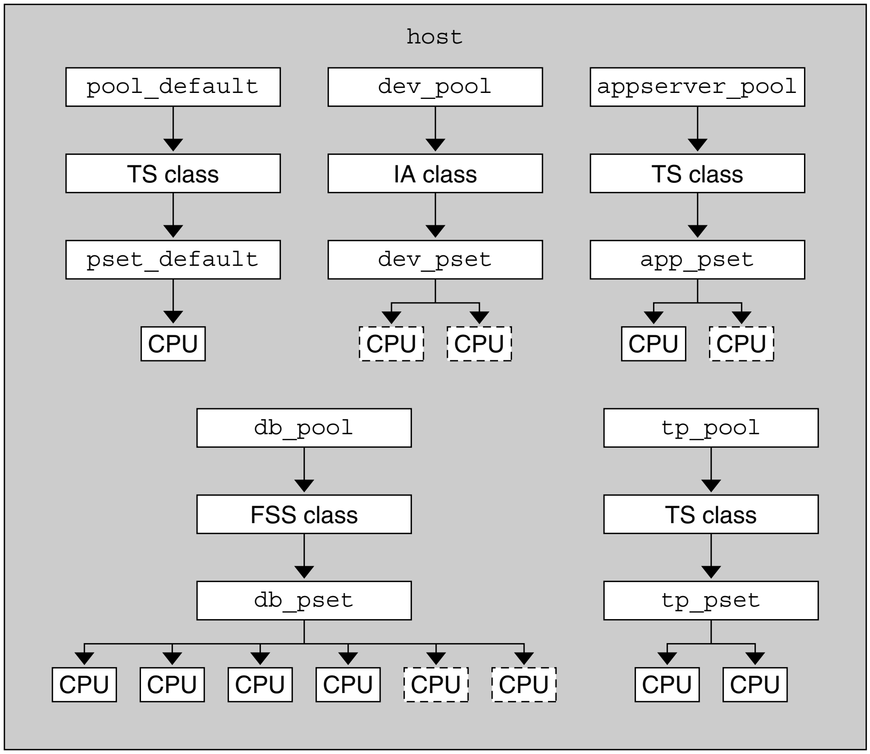
```
                boolean pset.default false
                uint    pset.min 1
                uint    pset.max 2
                string  pset.comment
                boolean pset.escapable false
                uint    pset.load 0
                uint    pset.size 0
                string  pset.poold.objectives locality tight; utilization < 80
                cpu
                        int     cpu.sys_id 7
                        string  cpu.comment
                        string  cpu.status on-line

        pset pset_default
                int     pset.sys_id -1
                string  pset.units population
                boolean pset.default true
                uint    pset.min 1
                uint    pset.max 4294967295
                string  pset.comment
                boolean pset.escapable false
                uint    pset.load 0
                uint    pset.size 0

                cpu
                        int     cpu.sys_id 0
                        string  cpu.comment
                        string  cpu.status on-line
```

A graphic representation of the framework follows.

FIGURE 14–1 Server Consolidation Configuration



Note – In the pool db_pool, the standalone database instance is guaranteed 75 percent of the CPU resource.

P A R T I I

Oracle Solaris Zones

This part covers Oracle Solaris Zones software partitioning technology, which provides a means of virtualizing operating system services to create an isolated environment for running applications. This isolation prevents processes that are running in one zone from monitoring or affecting processes running in other zones.

CHAPTER 15

Introduction to Oracle Solaris Zones

The Oracle Solaris Zones facility in the Oracle Solaris operating system provides an isolated environment in which to run applications on your system.

This chapter provides an overview of zones.

The chapter also covers the following general zones topics:

If you are ready to start creating zones on your system, skip to Chapter 16, "Non-Global Zone Configuration (Overview)."

Note – For information about Oracle Solaris 10 Zones, see Part III, "Oracle Solaris 10 Zones."

For information on using zones on an Oracle Solaris Trusted Extensions system, see Chapter 16, "Managing Zones in Trusted Extensions (Tasks)," in *Oracle Solaris Trusted Extensions Configuration and Administration*.

Zones Overview

The Oracle Solaris Zones partitioning technology is used to virtualize operating system services and provide an isolated and secure environment for running applications. A *zone* is a virtualized operating system environment created within a single instance of the Oracle Solaris operating system. When you create a zone, you produce an application execution environment in which processes are isolated from the rest of the system. This isolation prevents processes that are running in one zone from monitoring or affecting processes that are running in other zones. Even a process running with superuser credentials cannot view or affect activity in other zones. With Oracle Solaris Zones, you can maintain the one-application-per-server deployment model while simultaneously sharing hardware resources.

A zone also provides an abstract layer that separates applications from the physical attributes of the machine on which they are deployed. Examples of these attributes include physical device paths.

Zones can be used on any machine that is running the Oracle Solaris 10 or later Oracle Solaris release. The upper limit for the number of zones on a system is 8192. The number of zones that can be effectively hosted on a single system is determined by the total resource requirements of the application software running in all of the zones, and the size of the system.

The zone root file system model supported on this release is the *whole root zone*. This model provides the maximum configurability. All of the required system software and any additional packages are installed into the private file systems of the zone.

These concepts are discussed in Chapter 17, "Planning and Configuring Non-Global Zones (Tasks)."

About Oracle Solaris Zones in This Release

The default zone brand in the Oracle Solaris 11 Express release is the `ipkg` brand, described in this guide and in the `ipkg(5)` man page.

The `ipkg` branded zone is supported on all `sun4u`, `sun4v`, and x86 architecture machines.

The `ipkg` branded zone uses the branded zones framework described in the `brands(5)` man page to run zones installed with the same software as is installed in the global zone. The system software must always be in sync with the global zone when using an `ipkg` brand. The system software packages within the zone are managed using the Image Packaging System (IPS). IPS is the packaging system on the Oracle Solaris 11 release, and zones have changed to utilize this model.

The following differences between `ipkg` zones and `native` zones on earlier releases should be noted:

- The `ipkg` brand is the default instead of the `native` brand, which is the default on Oracle Solaris 10 systems.

- ipkg branded zones are whole-root type only.

  The sparse root type of native zone available on Oracle Solaris 10 uses the SVR4 package management system, and IPS doesn't use this framework.
- Zones in this release have software management related functionality that is different from the Oracle Solaris 10 release in these areas:
  - IPS versus SVR4 packaging
  - Install, detach/attach, and "physical to virtual" (P2V) capability
- Currently, zones employ manual syncing. The zones do not automatically update when you `pkg image-update` the system. You must manually update the zones after rebooting to keep them in sync with the global zone. See "Updating Non-Global Zones in the Oracle Solaris 11 Express Release" on page 357 for the workaround.
- Global zones are integrated with `beadm` and use boot environments.
- The non-global zone root is a ZFS dataset.
- You must be on the network to install a zone. See *Oracle Solaris 11 Express Image Packaging System Guide* for more information.
- Zone software is minimized to start. Any additional packages the zone requires must be added. See the solaris publisher (`http://pkg.oracle.com/solaris/release/`) for more information.
- To install additional software in the `ipkg` brand zone during initial installation, use the `-e` option to the `zoneadm install` command. See "How to Install a Configured Zone" on page 265 for an example command line.
- To achieve a full rather than a minimized zone, you can install the approximately 9.5–gigabyte redistributable incorporation or the approximately 3–gigabyte slim_install incorporation by using the `-e` option. See "How Zones Are Installed" on page 255 and "How to Install a Configured Zone" on page 265 for more information.

About Branded Zones

By default, a non-global zone on a system runs the same operating system software as the global zone. The branded zone (BrandZ) facility in the Oracle Solaris operating system is a simple extension of Oracle Solaris Zones. The BrandZ framework is used to create non-global branded zones that contain operating environments that are different from that of the global zone. Branded zones are used on the Oracle Solaris operating system to run applications. The BrandZ framework extends the Oracle Solaris Zones infrastructure in a variety of ways. These extensions can be complex, such as providing the capability to run different operating system environments within the zone, or simple, such as enhancing the base zone commands to provide new capabilities. For example, the Oracle Solaris 10 Container is a branded non-global zone that can emulate the Oracle Solaris 10 operating system. Even default zones that share the same operating system as the global zone are configured with a *brand*.

The brand defines the operating environment that can be installed in the zone, and determines how the system will behave within the zone so that the software installed in the zone functions correctly. In addition, a zone's brand is used to identify the correct application type at application launch time. All branded zone management is performed through extensions to the standard zones structure. Most administration procedures are identical for all zones.

The resources included in the configuration by default, such as defined file systems and privileges, are covered in the documentation for the brand.

BrandZ extends the zones tools in the following ways:

- The `zonecfg` command is used to set a zone's brand type when the zone is configured.
- The `zoneadm` command is used to report a zone's brand type as well as administer the zone.

Although you can configure and install branded zones on an Oracle Solaris Trusted Extensions system that has labels enabled, you cannot boot branded zones on this system configuration, *unless* the brand being booted is the `labeled` brand on a certified system configuration.

You can change the brand of a zone in the *configured* state. Once a branded zone has been *installed*, the brand cannot be changed or removed.

Caution – If you plan to migrate your existing Oracle Solaris 10 system into a `solaris10` branded zone on a system running the Oracle Solaris 11 Express release, you must migrate any existing zones to the target system first. Because zones do not nest, the system migration process renders any existing zones unusable. See Part III, "Oracle Solaris 10 Zones," for more information.

Processes Running in a Branded Zone

Branded zones provide a set of interposition points in the kernel that are only applied to processes executing in a branded zone.

- These points are found in such paths as the `syscall` path, the process loading path, and the thread creation path.
- At each of these points, a brand can choose to supplement or replace the standard Oracle Solaris behavior.

A brand can also provide a plug-in library for `librtld_db`. The plug-in library allows Oracle Solaris tools such as the debugger, described in `mdb(1)`, and DTrace, described in `dtrace(1M)`, to access the symbol information of processes running inside a branded zone.

Branded Zones Available in this Release

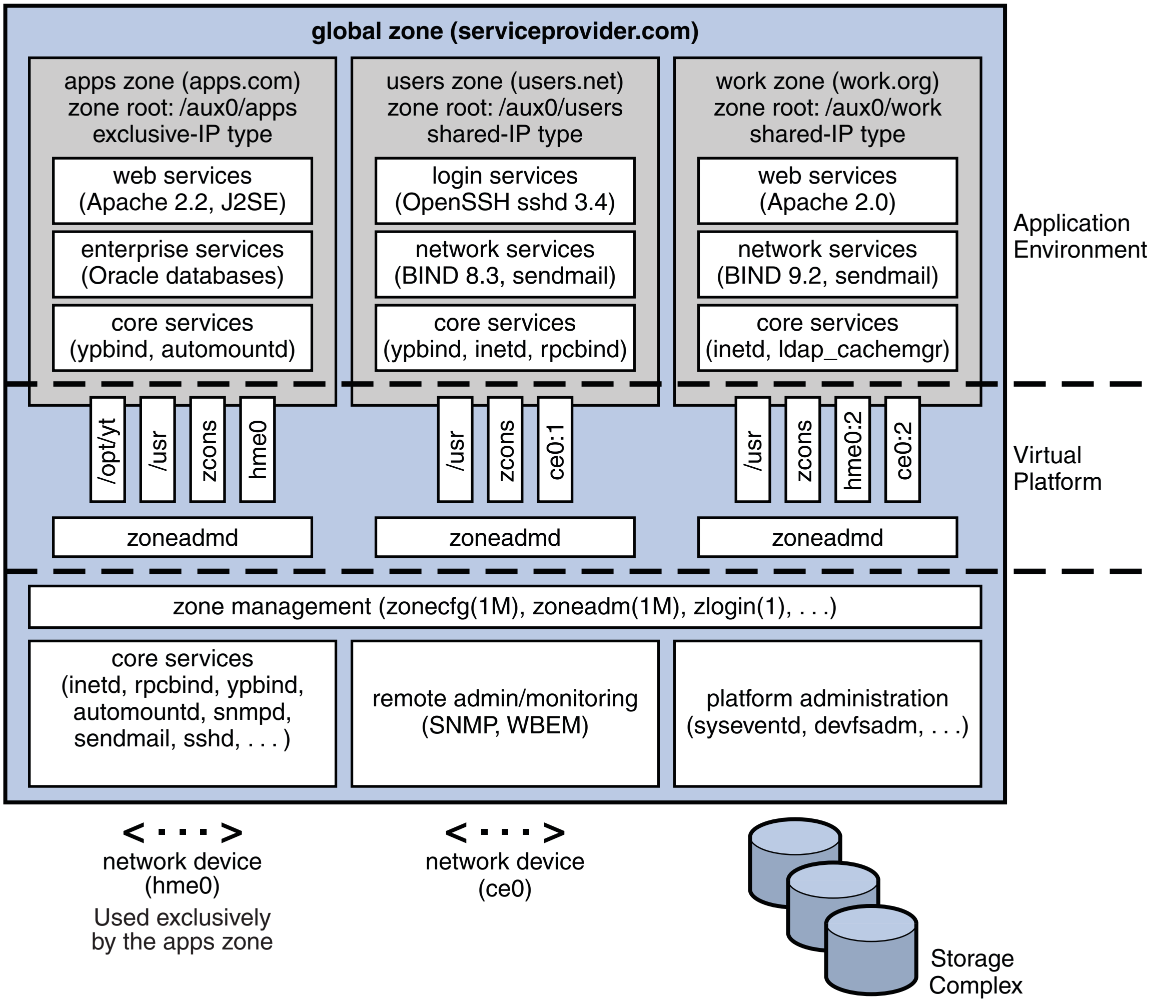
In addition to the default `ipkg` brand, Oracle Solaris 10 Zones (`solaris10` branded zones) is included in this release. For more information, see Part III, "Oracle Solaris 10 Zones."

When to Use Zones

Zones are ideal for environments that consolidate a number of applications on a single server. The cost and complexity of managing numerous machines make it advantageous to consolidate several applications on larger, more scalable servers.

The following figure shows a system with four zones. Each of the zones `apps`, `users`, and `work` is running a workload unrelated to the workloads of the other zones, in a sample consolidated environment. This example illustrates that different versions of the same application can be run without negative consequences in different zones, to match the consolidation requirements. Each zone can provide a customized set of services.

FIGURE 15–1 Zones Server Consolidation Example



Zones enable more efficient resource utilization on your system. Dynamic resource reallocation permits unused resources to be shifted to other zones as needed. Fault and security isolation mean that poorly behaved applications do not require a dedicated and under-utilized system. With the use of zones, these applications can be consolidated with other applications.

Zones allow you to delegate some administrative functions while maintaining overall system security.

How Zones Work

A non-global zone can be thought of as a box. One or more applications can run in this box without interacting with the rest of the system. Oracle Solaris zones isolate software applications or services by using flexible, software-defined boundaries. Applications that are running in the same instance of the Oracle Solaris operating system can then be managed independently of one other. Thus, different versions of the same application can be run in different zones, to match the requirements of your configuration.

A process assigned to a zone can manipulate, monitor, and directly communicate with other processes that are assigned to the same zone. The process cannot perform these functions with processes that are assigned to other zones in the system or with processes that are not assigned to a zone. Processes that are assigned to different zones are only able to communicate through network APIs.

IP networking can be configured in two different ways, depending on whether the zone has its own exclusive IP instance or shares the IP layer configuration and state with the global zone. For more information about IP types in zones, see "Zone Network Interfaces" on page 210. For configuration information, see "How to Configure the Zone" on page 238.

Every Oracle Solaris system contains a *global zone*. The global zone has a dual function. The global zone is both the default zone for the system and the zone used for system-wide administrative control. All processes run in the global zone if no *non-global* zones, referred to simply as zones, are created by the *global administrator* or a user with the Zone Security profile.

The global zone is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled. Only the global zone is bootable from the system hardware. Administration of the system infrastructure, such as physical devices, routing in a shared-IP zone, or dynamic reconfiguration (DR), is only possible in the global zone. Appropriately privileged processes running in the global zone can access objects associated with other zones.

Unprivileged processes in the global zone might be able to perform operations not allowed to privileged processes in a non-global zone. For example, users in the global zone can view information about every process in the system. If this capability presents a problem for your site, you can restrict access to the global zone.

Each zone, including the global zone, is assigned a zone name. The global zone always has the name `global`. Each zone is also given a unique numeric identifier, which is assigned by the system when the zone is booted. The global zone is always mapped to ID `0`. Zone names and numeric IDs are discussed in "Using the `zonecfg` Command" on page 218.

Each zone also has a node name that is completely independent of the zone name. The node name is assigned by the administrator of the zone. For more information, see "Non-Global Zone Node Name" on page 309.

Each zone has a path to its root directory that is relative to the global zone's root directory. For more information, see "Using the `zonecfg` Command" on page 218.

The scheduling class for a non-global zone is set to the scheduling class for the system by default. See “Scheduling Class” on page 208 for a discussion of methods used to set the scheduling class in a zone.

Summary of Oracle Solaris Zone Features

The following table summarizes the characteristics of global and non-global zones.

Type of Zone	Characteristic
Global	■ Is assigned ID `0` by the system ■ Provides the single instance of the Oracle Solaris kernel that is bootable and running on the system ■ Contains a complete installation of the Oracle Solaris system software packages ■ Can contain additional software packages or additional software, directories, files, and other data not installed through packages ■ Provides a complete and consistent product database that contains information about all software components installed in the global zone ■ Holds configuration information specific to the global zone only, such as the global zone host name and file system table ■ Is the only zone that is aware of all devices and all file systems ■ Is the only zone with knowledge of non-global zone existence and configuration ■ Is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled

Type of Zone	Characteristic
Non-Global	■ Is assigned a zone ID by the system when the zone is booted ■ Shares operation under the Oracle Solaris kernel booted from the global zone ■ Contains an installed subset of the complete Oracle Solaris operating system software packages ■ Can contain additional installed software packages ■ Can contain additional software, directories, files, and other data created on the non-global zone that are not installed through packages ■ Has a complete and consistent product database that contains information about all software components installed on the zone ■ Is not aware of the existence of any other zones ■ Cannot install, manage, or uninstall other zones, including itself ■ Has configuration information specific to that non-global zone only, such as the non-global zone host name and file system table ■ Can have its own time zone setting

How Non-Global Zones Are Administered

A global administrator has superuser privileges or equivalent rights profile. When logged in to the global zone, the global administrator can monitor and control the system as a whole.

A non-global zone can be administered by a *zone administrator*. The global administrator assigns the required authorizations to the zone administrator as described in "`admin` Resource" on page 206. The privileges of a zone administrator are confined to a non-global zone.

How Non-Global Zones Are Created

The global administrator or a user granted the Zone Security profile uses the `zonecfg` command to configure a zone by specifying various parameters for the zone's virtual platform and application environment. The zone is then installed by the global administrator, who uses the zone administration command `zoneadm` to install software at the package level into the file system hierarchy established for the zone. The `zoneadm` command is used to boot the zone. The global administrator or authorized user can then log in to the installed zone by using the `zlogin` command. If role-based access control (RBAC) is in use, the zone administrator must have the authorization `solaris.zone.manage/`*zonename*.

At first login, the internal configuration for the zone is completed.

For information about zone configuration, see Chapter 16, "Non-Global Zone Configuration (Overview)." For information about zone installation, see Chapter 18, "About Installing, Halting, Uninstalling, and Cloning Non-Global Zones (Overview)." For information about zone login, see Chapter 20, "Non-Global Zone Login (Overview)."

Non-Global Zone State Model

A non-global zone can be in one of the following six states:

Configured — The zone's configuration is complete and committed to stable storage. However, those elements of the zone's application environment that must be specified after initial boot are not yet present.

Incomplete — During an install or uninstall operation, `zoneadm` sets the state of the target zone to incomplete. Upon successful completion of the operation, the state is set to the correct state.

A damaged installed zone can be marked incomplete by using the `mark` subcommand of `zoneadm`. Zones in the incomplete state are shown in the output of `zoneadm list -iv`.

Installed — The zone's configuration is instantiated on the system. The `zoneadm` command is used to verify that the configuration can be successfully used on the designated Oracle Solaris system. Packages are installed under the zone's root path. In this state, the zone has no associated virtual platform.

Ready — The virtual platform for the zone is established. The kernel creates the `zsched` process, network interfaces are set up and made available to the zone, file systems are mounted, and devices are configured. A unique zone ID is assigned by the system. At this stage, no processes associated with the zone have been started.

Running — User processes associated with the zone application environment are running. The zone enters the running state as soon as the first user process associated with the application environment (`init`) is created.

Shutting down and Down — These states are transitional states that are visible while the zone is being halted. However, a zone that is unable to shut down for any reason will stop in one of these states.

Chapter 19, "Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks)," and the `zoneadm(1M)` man page describe how to use the `zoneadm` command to initiate transitions between these states.

TABLE 15–1 Commands That Affect Zone State

Current Zone State	Applicable Commands

Configured	`zonecfg -z` *zonename* `verify`
	
	`zonecfg -z` *zonename* `commit`
	
	`zonecfg -z` *zonename* `delete`
	
	`zoneadm -z` *zonename* `attach`
	
	`zoneadm -z` *zonename* `verify`
	
	`zoneadm -z` *zonename* `install`
	
	`zoneadm -z` *zonename* `clone`
	
	You can also use `zonecfg` to rename a zone in the configured or installed state.

Incomplete	`zoneadm -z` *zonename* `uninstall`

Installed	`zoneadm -z` *zonename* `ready` (optional)
	
	`zoneadm -z` *zonename* `boot`
	
	`zoneadm -z` *zonename* `uninstall` uninstalls the configuration of the specified zone from the system.
	
	`zoneadm -z` *zonename* `move` *path*
	
	`zoneadm -z` *zonename* `detach`
	
	`zonecfg -z` *zonename* can be used to add or remove an `attr`, `bootargs`, `capped-memory`, `dataset`, `capped-cpu`, `dedicated-cpu`, `device`, `fs`, `ip-type`, `limitpriv`, `net`, `rctl`, or `scheduling-class` property. You can also rename a zone in the installed state.

Ready	`zoneadm -z` *zonename* `boot` `zoneadm halt` and system reboot return a zone in the ready state to the installed state. `zonecfg -z` *zonename* can be used to add or remove `attr`, `bootargs`, `capped-memory`, `dataset`, `capped-cpu`, `dedicated-cpu`, `device`, `fs`, `ip-type`, `limitpriv`, `net`, `rctl`, or `scheduling-class` property.

TABLE 15–1 Commands That Affect Zone State *(Continued)*

Current Zone State	Applicable Commands

Running	`zlogin` *options* `zonename`
	
	`zoneadm -z` *zonename* `reboot`
	
	`zoneadm -z` *zonename* `halt` returns a ready zone to the installed state.
	
	`zoneadm halt` and system reboot return a zone in the running state to the installed state.
	
	`zonecfg -z` *zonename* can be used to add or remove an `attr`, `bootargs`, `capped-memory`, `dataset`, `capped-cpu`, `dedicated-cpu`, `device`, `fs`, `ip-type`, `limitpriv`, `net`, `rctl`, or `scheduling-class` property. The `zonepath` resource cannot be changed.

Note – Parameters changed through `zonecfg` do not affect a running zone. The zone must be rebooted for the changes to take effect.

Non-Global Zone Characteristics

A zone provides isolation at almost any level of granularity you require. A zone does not need a dedicated CPU, a physical device, or a portion of physical memory. These resources can either be multiplexed across a number of zones running within a single domain or system, or allocated on a per-zone basis using the resource management features available in the operating system.

Each zone can provide a customized set of services. To enforce basic process isolation, a process can see or signal only those processes that exist in the same zone. Basic communication between zones is accomplished by giving each zone IP network connectivity. An application running in one zone cannot observe the network traffic of another zone. This isolation is maintained even though the respective streams of packets travel through the same physical interface.

Each zone is given a portion of the file system hierarchy. Because each zone is confined to its subtree of the file system hierarchy, a workload running in a particular zone cannot access the on-disk data of another workload running in a different zone.

Files used by naming services reside within a zone's own root file system view. Thus, naming services in different zones are isolated from one other and the services can be configured differently.

Using Resource Management Features With Non-Global Zones

If you use resource management features, you should align the boundaries of the resource management controls with those of the zones. This alignment creates a more complete model of a virtual machine, where namespace access, security isolation, and resource usage are all controlled.

Any special requirements for using the various resource management features with zones are addressed in the individual chapters of this manual that document those features.

Monitoring Non-Global Zones

To report on the CPU, memory, and resource control utilization of the currently running zones, see "Using the `zonestat` Utility in a Non-Global Zone" on page 341.

Features Provided by Non-Global Zones

Non-global zones provide the following features:

Security — Once a process has been placed in a zone other than the global zone, neither the process nor any of its subsequent children can change zones.

Network services can be run in a zone. By running network services in a zone, you limit the damage possible in the event of a security violation. An intruder who successfully exploits a security flaw in software running within a zone is confined to the restricted set of actions possible within that zone. The privileges available within a zone are a subset of those available in the system as a whole.

Isolation — Zones allow the deployment of multiple applications on the same machine, even if those applications operate in different trust domains, require exclusive access to a global resource, or present difficulties with global configurations. For example, multiple applications running in different shared-IP zones on the same system can bind to the same network port by using the distinct IP addresses associated with each zone or by using the wildcard address. The applications are also prevented from monitoring or intercepting each other's network traffic, file system data, or process activity.

Network Isolation — If a zone needs to be isolated at the IP layer on the network, for example, by being connected to different VLANs or different LANs than the global zone and other non-global zones, then for security reasons the zone can have an exclusive IP. The exclusive-IP zone can be used to consolidate applications that must communicate on different subnets that are on different VLANs or different LANs.

Zones can also be configured as shared-IP zones. These zones connect to the same VLANs or same LANs as the global zone and share the IP routing configuration with the global zone. Shared-IP zones have separate IP addresses, but share the other parts of IP.

Virtualization — Zones provide a virtualized environment that can hide details such as physical devices and the system's primary IP address and host name from applications. The same application environment can be maintained on different physical machines. The virtualized environment allows separate administration of each zone. Actions taken by a zone administrator in a non-global zone do not affect the rest of the system.

Granularity — A zone can provide isolation at almost any level of granularity. See "Non-Global Zone Characteristics" on page 200 for more information.

Environment — Zones do not change the environment in which applications execute except when necessary to achieve the goals of security and isolation. Zones do not present a new API or ABI to which applications must be ported. Instead, zones provide the standard Oracle Solaris interfaces and application environment, with some restrictions. The restrictions primarily affect applications that attempt to perform privileged operations.

Applications in the global zone run without modification, whether or not additional zones are configured.

Setting Up Zones on Your System (Task Map)

The following table provides a basic overview of the tasks that are involved in setting up zones on your system for the first time.

Task	Description	For Instructions
Identify the applications that you would like to run in zones.	Review the applications running on your system: ■ Determine which applications are critical to your business goals. ■ Assess the system needs of the applications you are running.	Refer to your business goals and to your system documentation if necessary.

Task	Description	For Instructions
Determine how many zones to configure.	Assess: ■ The performance requirements of the applications you intend to run in zones. ■ The availability of 256 megabytes to 5 gigabytes of disk space per zone to be installed. The amount required is dependent on the software to be installed inside the zone.	See "Evaluating the Current System Setup" on page 234.
Determine whether your zone will use resource pools or assigned CPUs to partition machine resources.	If you are also using resource management features on your system, align the zones with the resource management boundaries. Configure resource pools before you configure zones. Note that you can add zone-wide resource controls and pool functionality to a zone quickly by using `zonecfg` properties.	See "How to Configure the Zone" on page 238, and Chapter 13, "Creating and Administering Resource Pools (Tasks)."
Perform the preconfiguration tasks.	Determine the zone name and the zone path. Determine whether the zone will be a shared-IP zone or an exclusive-IP zone, and obtain IP addresses or the data-link name. Determine the required file systems and devices for each zone. Determine the scheduling class for the zone. Determine the set of privileges that processes inside the zone should be limited to, if the standard default set is not sufficient. Note that some `zonecfg` settings automatically add privileges. For example, `ip-type=exclusive` automatically adds multiple privileges required to configure and manage network stacks.	For information on the zone name and path, IP types, IP addresses, file systems, devices, scheduling class, and privileges, see Chapter 16, "Non-Global Zone Configuration (Overview)," and "Evaluating the Current System Setup" on page 234. For a listing of default privileges and privileges that can be configured in a non-global zone, see "Privileges in a Non-Global Zone" on page 324. For information about IP feature availability, see "Networking in Shared-IP Non-Global Zones" on page 316 and "Networking in Exclusive-IP Non-Global Zones" on page 319.

Task	Description	For Instructions
Develop configurations.	Configure non-global zones.	See "Configuring, Verifying, and Committing a Zone" on page 238 and the zonecfg(1M) man page.
As global administrator or a user with appropriate authorizations, verify and install configured zones.	Zones must be verified and installed prior to login.	See Chapter 18, "About Installing, Halting, Uninstalling, and Cloning Non-Global Zones (Overview)," and Chapter 19, "Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks)."
As global administrator or a user granted appropriate authorizations, boot the non-global zones.	Boot each zone to place the zone in the running state.	See Chapter 18, "About Installing, Halting, Uninstalling, and Cloning Non-Global Zones (Overview)," and Chapter 19, "Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks)."
As global administrator or a user granted appropriate authorizations, perform the initial internal configuration of the zone.	Place a `sysidcfg` file in the zone's `/etc` directory, or log in to each non-global zone using the `zlogin` command with the `-C` option and enter the requested information, including assigning the zone root password.	See Chapter 20, "Non-Global Zone Login (Overview)," and Chapter 21, "Logging In to Non-Global Zones (Tasks)."
Prepare the new zone for production use.	Create user accounts, add additional software, and customize the zone's configuration.	Refer to the documentation you use to set up a newly installed machine. Special considerations applicable to a system with zones installed are covered in this guide.

CHAPTER 16

Non-Global Zone Configuration (Overview)

This chapter provides an introduction to non-global zone configuration.

The following topics are covered in this chapter:

After you have learned about zone configuration, go to Chapter 17, "Planning and Configuring Non-Global Zones (Tasks)," to configure non-global zones for installation on your system.

About Resources in Zones

Resources that can be controlled in a zone include the following:

- Resource pools or assigned CPUs, which are used for partitioning machine resources.
- Resource controls, which provide a mechanism for the constraint of system resources.
- Scheduling class, which enables you to control the allocation of available CPU resources among zones, based on their importance. This importance is expressed by the number of shares of CPU resources that you assign to each zone.

Pre-Installation Configuration Process

Before you can install a non-global zone and use it on your system, the zone must be configured.

The zonecfg command is used to create the configuration and to determine whether the specified resources and properties are valid on a hypothetical system. The check performed by zonecfg for a given configuration verifies the following:

- Ensures that a zone path is specified
- Ensures that all of the required properties for each resource are specified

For more information about the zonecfg command, see the zonecfg(1M) man page.

Zone Components

This section covers the required and optional zone components that can be configured. Only the zone name and zone path are required. Additional information is provided in "Zone Configuration Data" on page 221.

Zone Name and Path

You must choose a name and a path for your zone. The zone must reside on a ZFS dataset. The ZFS dataset will be created automatically when the zone is installed or attached. If a ZFS dataset cannot be created, the zone will not install or attach. Note that the parent directory of the zone path must also be a dataset.

Zone Autoboot

The autoboot property setting determines whether the zone is automatically booted when the global zone is booted. The zones service, svc:/system/zones:default must also be enabled.

Also note that if the zone is set autoboot=true, then this setting should be changed to autoboot=false when a pkg image-update is performed. See "Zones Packaging Overview" on page 301. Once the new BE is booted and the zones are synced up to the global zone, then autoboot can be turned back on (reset to true).

admin Resource

The admin setting allows you to set zone administration authorization. The preferred method for defining authorizations is through the zonecfg command.

user Specify the user name.

`auths`	Specify the authorizations for the user name.	
	`solaris.zone.login`	If role-based access control (RBAC) is in use, the authorization `solaris.zone.login/`*zonename*is required for interactive logins. Password authentication takes place in the zone.
	`solaris.zone.manage`	If RBAC is in use, for non-interactive logins, or to bypass password authentication, the authorization `solaris.zone.manage/`*zonename* is required.
	`solaris.zone.clonefrom`	If RBAC is in use, subcommands that make a copy of another zone require the authorization, `solaris.zone.clonefrom/`*source_zone*.

Resource Pool Association

If you have configured resource pools on your system as described in Chapter 13, "Creating and Administering Resource Pools (Tasks)," you can use the `pool` property to associate the zone with one of the resource pools when you configure the zone.

If you do not have resource pools configured, you can still specify that a subset of the system's processors be dedicated to a non-global zone while it is running by using the `dedicated-cpu` resource. The system will dynamically create a temporary pool for use while the zone is running. With specification through `zonecfg`, pool settings propagate during migrations.

Note – A zone configuration using a persistent pool set through the `pool` property is incompatible with a temporary pool configured through the `dedicated-cpu` resource. You can set only one of these two properties.

`dedicated-cpu` Resource

The `dedicated-cpu` resource specifies that a subset of the system's processors should be dedicated to a non-global zone while it is running. When the zone boots, the system will dynamically create a temporary pool for use while the zone is running.

With specification in `zonecfg`, pool settings propagate during migrations.

The `dedicated-cpu` resource sets limits for `ncpus`, and optionally, `importance`.

`ncpus` Specify the number of CPUs or specify a range, such as 2–4 CPUs. If you specify a range because you want dynamic resource pool behavior, also do the following:

- Set the `importance` property.
- Enable the `poold` service. For instructions, see "How to Enable the Dynamic Resource Pools Service Using svcadm" on page 160.

`importance` — If you are using a CPU range to achieve dynamic behavior, also set the `importance` property, The `importance` property, which is *optional*, defines the relative importance of the pool. This property is only needed when you specify a range for `ncpus` and are using dynamic resource pools managed by `poold`. If `poold` is not running, then `importance` is ignored. If `poold` is running and `importance` is not set, `importance` defaults to `1`. For more information, see "pool.importance Property Constraint" on page 145.

Note – The `capped-cpu` resource and the `dedicated-cpu` resource are incompatible. The `cpu-shares` rctl and the `dedicated-cpu` resource are incompatible.

---

capped-cpu Resource

The `capped-cpu` resource provides an absolute fine-grained limit on the amount of CPU resources that can be consumed by a project or a zone. When used in conjunction with processor sets, CPU caps limit CPU usage within a set. The `capped-cpu` resource has a single `ncpus` property that is a positive decimal with two digits to the right of the decimal. This property corresponds to units of CPUs. The resource does not accept a range. The resource does accept a decimal number. When specifying `ncpus`, a value of `1` means 100 percent of a CPU. A value of `1.25` means 125 percent, because 100 percent corresponds to one full CPU on the system.

Note – The `capped-cpu` resource and the `dedicated-cpu` resource are incompatible.

Scheduling Class

You can use the *fair share scheduler* (FSS) to control the allocation of available CPU resources among zones, based on their importance. This importance is expressed by the number of *shares* of CPU resources that you assign to each zone. Even if you are not using FSS to manage CPU resource allocation between zones, you can set the zone's scheduling-class to use FSS so that you can set shares on projects within the zone.

When you explicitly set the `cpu-shares` property, the fair share scheduler (FSS) will be used as the scheduling class for that zone. However, the preferred way to use FSS in this case is to set FSS to be the system default scheduling class with the `dispadmin` command. That way, all zones will benefit from getting a fair share of the system CPU resources. If `cpu-shares` is not set for a zone, the zone will use the system default scheduling class. The following actions set the scheduling class for a zone:

- You can use the `scheduling-class` property in `zonecfg` to set the scheduling class for the zone.
- You can set the scheduling class for a zone through the resource pools facility. If the zone is associated with a pool that has its `pool.scheduler` property set to a valid scheduling class, then processes running in the zone run in that scheduling class by default. See "Introduction to Resource Pools" on page 136 and "How to Associate a Pool With a Scheduling Class" on page 166.
- If the `cpu-shares` rctl is set and FSS has not been set as the scheduling class for the zone through another action, `zoneadmd` sets the scheduling class to FSS when the zone boots.
- If the scheduling class is not set through any other action, the zone inherits the system default scheduling class.

Note that you can use the `priocntl` described in the `priocntl(1)` man page to move running processes into a different scheduling class without changing the default scheduling class and rebooting.

Physical Memory Control and the `capped-memory` Resource

The `capped-memory` resource sets limits for `physical`, `swap`, and `locked` memory. Each limit is optional, but at least one must be set.

- Determine values for this resource if you plan to cap memory for the zone by using `rcapd` from the global zone. The `physical` property of the `capped-memory` resource is used by `rcapd` as the `max-rss` value for the zone.
- The `swap` property of the `capped-memory` resource is the preferred way to set the `zone.max-swap` resource control.
- The `locked` property of the `capped-memory` resource is the preferred way to set the `zone.max-locked-memory` resource control.

Note – Applications generally do not lock significant amounts of memory, but you might decide to set locked memory if the zone's applications are known to lock memory. If zone trust is a concern, you can also consider setting the locked memory cap to 10 percent of the system's physical memory, or 10 percent of the zone's physical memory cap.

For more information, see Chapter 10, "Physical Memory Control Using the Resource Capping Daemon (Overview)," Chapter 11, "Administering the Resource Capping Daemon (Tasks)," and "How to Configure the Zone" on page 238. To temporarily set a resource cap for a zone, see "How to Specify a Temporary Resource Cap for a Zone" on page 131.

Zone Network Interfaces

Zone network interfaces configured by the `zonecfg` command to provide network connectivity will automatically be set up and placed in the zone when it is booted.

The Internet Protocol (IP) layer accepts and delivers packets for the network. This layer includes IP routing, the Address Resolution Protocol (ARP), IP security architecture (IPsec), and IP Filter.

There are two IP types available for non-global zones, shared-IP and exclusive-IP. The shared-IP zone shares a network interface and the exclusive-IP zone must have a dedicated network interface.

For information about IP features in each type, see "Networking in Shared-IP Non-Global Zones" on page 316 and "Networking in Exclusive-IP Non-Global Zones" on page 319.

Note – The link protection feature described in Chapter 18, "Using Link Protection in Virtualized Environments," in *System Administration Guide: Network Interfaces and Network Virtualization* can be used on a system running zones. This feature is configured in the global zone.

About Data-Links

A data-link is an interface at Layer 2 of the OSI protocol stack, which is represented in a system as a STREAMS DLPI (v2) interface. Such an interface can be plumbed under protocol stacks such as TCP/IP. Data-links are physical interfaces `e1000g0`, `bge3` (as NICs), `aggr1`, `aggr2` (as aggregations), or `e1000g123000`, `bge234003` (as VLAN 123 on `e1000g0` and VLAN 234 on `bge3`, respectively). physical interfaces (e.g. `e1000g0`, `bge1`), aggregations (`aggr3`), or VLAN-tagged interfaces (`e1000g111000` (VLAN tag 111 on `e1000g0`), `bge111001`, `aggr111003`). A data-link may also be referred to as a physical interface, such as when referring to a Network Interface Card (NIC). The data-link is the `physical` property configured with the zone configuration tool `zonecfg`(1M). The `physical` property can be a VNIC, as described in Part IV, " Network Virtualization and Resource Management," in *System Administration Guide: Network Interfaces and Network Virtualization*.

Data-links are `e1000g0`, `bge3` (as NICs), `aggr1`, `aggr2` (as aggregations), or `e1000g123000`, `bge234003` (as VLAN 123 on `e1000g0` and VLAN 234 on `bge3`, respectively).

Shared-IP Non-Global Zones

The shared-IP zone is the default type. The zone must have one or more dedicated IP addresses. A shared-IP zone shares the IP layer configuration and state with the global zone. The zone should use the shared-IP instance if both of the following are true:

- The non-global zone is to use the same data-link that is used by the global zone, regardless of whether the global and non-global zones are on the same subnet.
- You do not want the other capabilities that the exclusive-IP zone provides.

Shared-IP zones are assigned one or more IP addresses using the `zonecfg` command. The data-link names must also be configured in the global zone.

In the `zonecfg` `net` resource, the `address` and the `physical` properties must be set. The `defrouter` property is optional.

These addresses are associated with logical network interfaces. The `ifconfig` command can be used from the global zone to add or remove logical interfaces in a running zone. For more information, see "Shared-IP Network Interfaces" on page 317.

Exclusive-IP Non-Global Zones

Full IP-level functionality is available in an exclusive-IP zone.

An exclusive-IP zone has its own IP-related state.

This includes the ability to use the following features in an exclusive-IP zone:

- DHCPv4 and IPv6 stateless address autoconfiguration
- IP Filter, including network address translation (NAT) functionality
- IP Network Multipathing (IPMP)
- IP routing
- `ndd` for setting TCP/UDP/SCTP as well as IP/ARP-level knobs
- IP security (IPsec) and Internet Key Exchange (IKE), which automates the provision of authenticated keying material for IPsec security association

Query — add VNIC, (vdb1) net;set physical=vdb1;end An exclusive-IP zone is assigned its own set of data-links using the `zonecfg` command. The zone is given a data-link name such as `xge0`, `e1000g1`, or `bge32001`, using the `physical` property of the `net` resource. The `physical` property can be a VNIC, as described in Part IV, " Network Virtualization and Resource Management," in *System Administration Guide: Network Interfaces and Network Virtualization*. The `address` property of the `net` resource is not set.

The `defrouter` and the `allowed-addresses` properties of the `net` resource can optionally be set. Setting `allowed-addresses` allows only those addresses to be configured by the non-global zone. Data-link protection is enabled with the `allowed-addresses` property. When the non-global zone is booted:

- The IP interfaces are automatically created and configured.
- The default route is configured.

Note that the assigned data-link enables the `snoop` command to be used.

The `dladm` command can be used with the `show-linkprop` subcommand to show the assignment of data-links to running exclusive-IP zones. The `dladm` command can be used with the `set-linkprop` subcommand to assign additional data-links to running zones. See "Administering Data-Links in Exclusive-IP Non-Global Zones" on page 351 for usage examples.

Inside a running exclusive-IP zone, the `ifconfig` command can be used to configure IP, which includes the ability to add or remove logical interfaces. The IP configuration in a zone can be set up in the same way as for the global zone, by using the `sysidtools` described in `sysidcfg(4)`.

Note – The IP configuration of an exclusive-IP zone can only be viewed from the global zone by using the `zlogin` command. An example follows.

```
global# zlogin zone1 ifconfig -a
```

Security Differences Between Shared-IP and Exclusive-IP Non-Global Zones

In a shared-IP zone, applications in the zone, including the superuser, cannot send packets with source IP addresses other than the ones assigned to the zone through the `zonecfg` utility. This type of zone does not have access to send and receive arbitrary data-link (layer 2) packets.

For an exclusive-IP zone, `zonecfg` instead grants the entire specified data-link to the zone. As a result, in an exclusive-IP zone, the superuser or user with the required rights profile can send spoofed packets on those data-links, just as can be done in the global zone.

Using Shared-IP and Exclusive-IP Non-Global Zones at the Same Time

The shared-IP zones always share the IP layer with the global zone, and the exclusive-IP zones always have their own instance of the IP layer. Both shared-IP zones and exclusive-IP zones can be used on the same machine.

File Systems Mounted in Zones

Generally, the file systems mounted in a zone include the following:

- The set of file systems mounted when the virtual platform is initialized
- The set of file systems mounted from within the application environment itself

This can include, for example, the following file systems:

- File systems specified in a zone's `/etc/vfstab` file
- **Moved to SMF. Use sharectl(1M) to manage NFS properties. Get SMF replacement ?**

  `AutoFS` and `AutoFS`-triggered mounts
- Mounts explicitly performed by a zone administrator

  File system mounting permissions within a running zone are also defined by the `zonecfg` `fs-allowed` property. This property does not apply to file systems mounted into the zone by using the `zonecfg` `add fs` or `add dataset` resources. By default, only mounts of `hsfs` file systems, and network file systems such as NFS, are allowed within a zone.

Caution – Certain restrictions are placed on mounts other than the defaults performed from within the application environment. These restrictions prevent the zone administrator from denying service to the rest of the system, or otherwise negatively impacting other zones.

There are security restrictions associated with mounting certain file systems from within a zone. Other file systems exhibit special behavior when mounted in a zone. See “File Systems and Non-Global Zones” on page 310 for more information.

Host ID in Zones

You can set a `hostid` property for the non-global zone that is different from the `hostid` of the global zone. This would be done, for example, in the case of a machine migrated into a zone on another system. Applications now inside the zone might depend on the original `hostid`. See “Resource and Property Types” on page 221 for more information.

Configured Devices in Zones

The `zonecfg` command uses a rule-matching system to specify which devices should appear in a particular zone. Devices matching one of the rules are included in the zone's `/dev` file system. For more information, see “How to Configure the Zone” on page 238.

Disk Format Support in Non-Global Zones

Safe delegation of slices and disks to zones is enabled by using the `limitpriv` property described in "Resource and Property Types" on page 221:

- Delegation is only supported for `ipkg` zones
- Disks must use the `sd` target as shown by using the `prtconf` command with the `-D` option.
- If just a slice is being delegated, the following `limitpriv` must be set through the `zonecfg` utility:

  zonecfg:my-zone> **`set limitpriv="default,-storage_config"`**
- Raw, unsafe, SCSI access may be allowed by adding the `storage_raw` privilege.

Setting Zone-Wide Resource Controls

The global administrator or a user with appropriate authorizations can set privileged zone-wide resource controls for a zone. Zone-wide resource controls limit the total resource usage of all process entities within a zone.

These limits are specified for both the global and non-global zones by using the `zonecfg` command. See "How to Configure the Zone" on page 238.

The preferred, simpler method for setting a zone-wide resource control is to use the property name instead of the `rctl` resource.

The `zone.cpu-cap` resource control sets an absolute limit on the amount of CPU resources that can be consumed by a zone. A value of `100` means 100 percent of one CPU as the `project.cpu-cap` setting. A value of `125` is 125 percent, because 100 percent corresponds to one full CPU on the system when using CPU caps.

Note – When setting the `capped-cpu` resource, you can use a decimal number for the unit. The value correlates to the `zone.capped-cpu` resource control, but the setting is scaled down by 100. A setting of `1` is equivalent to a setting of `100` for the resource control.

The `zone.cpu-shares` resource control sets a limit on the number of fair share scheduler (FSS) CPU shares for a zone. CPU shares are first allocated to the zone, and then further subdivided among projects within the zone as specified in the `project.cpu-shares` entries. For more information, see "Using the Fair Share Scheduler on an Oracle Solaris System With Zones Installed" on page 353. The global property name for this control is `cpu-shares`.

The zone.max-locked-memory resource control limits the amount of locked physical memory available to a zone The allocation of the locked memory resource across projects within the zone can be controlled by using the project.max-locked-memory resource control. See Table 6–1 for more information.

The zone.max-lofi resource control limits the number of potential lofi devices that can be created by a zone.

The zone.max-lwps resource control enhances resource isolation by preventing too many LWPs in one zone from affecting other zones. The allocation of the LWP resource across projects within the zone can be controlled by using the project.max-lwps resource control. See Table 6–1 for more information. The global property name for this control is max-lwps.

The zone.max-processes resource control enhances resource isolation by preventing a zone from using too many process table slots and thus affecting other zones. The allocation of the process table slots resource across projects within the zone can be set by using the project.max-processes resource control described in "Available Resource Controls" on page 80. The global property name for this control is max-processes. The zone.max-processes resource control can also encompass the zone.max-lwps resource control. If zone.max-processes is set and zone.max-lwps is not set, then zone.max-lwps is implicitly set to 10 times the zone.max-processes value when the zone is booted. Note that because both normal processes and zombie processes take up process table slots, the max-processes control thus protects against zombies exhausting the process table. Because zombie processes do not have any LWPs by definition, the max-lwps cannot protect against this possibility.

The zone.max-msg-ids, zone.max-sem-ids, zone.max-shm-ids, and zone.max-shm-memory resource controls are used to limit System V resources used by all processes within a zone. The allocation of System V resources across projects within the zone can be controlled by using the project versions of these resource controls. The global property names for these controls are max-msg-ids, max-sem-ids, max-shm-ids, and max-shm-memory.

The zone.max-swap resource control limits swap consumed by user process address space mappings and tmpfs mounts within a zone. The output of prstat -Z displays a SWAP column. The swap reported is the total swap consumed by the zone's processes and tmpfs mounts. This value assists in monitoring the swap reserved by each zone, which can be used to choose an appropriate zone.max-swap setting.

TABLE 16–1 Zone-Wide Resource Controls

Control Name	Global Property Name	Description	Default Unit	Value Used For
`zone.cpu-cap`		Absolute limit on the amount of CPU resources for this zone	Quantity (number of CPUs), expressed as a percentage **Note –** When setting as the `capped-cpu` resource, you can use a decimal number for the unit.	
`zone.cpu-shares`	`cpu-shares`	Number of fair share scheduler (FSS) CPU shares for this zone	Quantity (shares)	
`zone.max-locked-memory`		Total amount of physical locked memory available to a zone. If `priv_proc_lock_memory` is assigned to a zone, consider setting this resource control as well, to prevent that zone from locking all memory.	Size (bytes)	`locked` property of `capped-memory`
`zone.max-lofi`	`max-lofi`	Limit on the number of potential `lofi` devices that can be created by a zone	Quantity (number of `lofi` devices)	
`zone.max-lwps`	`max-lwps`	Maximum number of LWPs simultaneously available to this zone	Quantity (LWPs)	
`zone.max-msg-ids`	`max-msg-ids`	Maximum number of message queue IDs allowed for this zone	Quantity (message queue IDs)	

TABLE 16–1 Zone-Wide Resource Controls *(Continued)*

Control Name	Global Property Name	Description	Default Unit	Value Used For
`zone.max-processes`	`max-processes`	Maximum number of process table slots simultaneously available to this zone	Quantity (process table slots)	
`zone.max-sem-ids`	`max-sem-ids`	Maximum number of semaphore IDs allowed for this zone	Quantity (semaphore IDs)	
`zone.max-shm-ids`	`max-shm-ids`	Maximum number of shared memory IDs allowed for this zone	Quantity (shared memory IDs)	
`zone.max-shm-memory`	`max-shm-memory`	Total amount of System V shared memory allowed for this zone	Size (bytes)	
`zone.max-swap`		Total amount of swap that can be consumed by user process address space mappings and `tmpfs` mounts for this zone.	Size (bytes)	`swap` property of `capped-memory`

These limits can be specified for running processes by using the `prctl` command. An example is provided in "How to Set FSS Shares in the Global Zone Using the `prctl` Command" on page 353. Limits specified through the `prctl` command are not persistent. The limits are only in effect until the system is rebooted.

Configurable Privileges

When a zone is booted, a default set of *safe* privileges is included in the configuration. These privileges are considered safe because they prevent a privileged process in the zone from affecting processes in other non-global zones on the system or in the global zone. You can use the `zonecfg` command to do the following:

- Add to the default set of privileges, understanding that such changes might allow processes in one zone to affect processes in other zones by being able to control a global resource.
- Remove from the default set of privileges, understanding that such changes might prevent some processes from operating correctly if they require those privileges to run.

Note – There are a few privileges that cannot be removed from the zone's default privilege set, and there are also a few privileges that cannot be added to the set at this time.

For more information, see "Privileges in a Non-Global Zone" on page 324, "How to Configure the Zone" on page 238, and `privileges(5)`.

Including a Comment for a Zone

You can add a comment for a zone by using the `attr` resource type. For more information, see "How to Configure the Zone" on page 238.

Using the `zonecfg` Command

The `zonecfg` command, which is described in the `zonecfg`(1M) man page, is used to configure a non-global zone.

The `zonecfg` command can also be used to persistently specify the resource management settings for the global zone. For example, you can use the command to configure the global zone to use a dedicated CPU by using the `dedicated-cpu` resource.

The `zonecfg` command can be used in interactive mode, in command-line mode, or in command-file mode. The following operations can be performed using this command:

- Create or delete (destroy) a zone configuration
- Add resources to a particular configuration
- Set properties for resources added to a configuration
- Remove resources from a particular configuration
- Query or verify a configuration
- Commit to a configuration
- Revert to a previous configuration
- Rename a zone
- Exit from a `zonecfg` session

The `zonecfg` prompt is of the following form:

```
zonecfg:zonename>
```

When you are configuring a specific resource type, such as a file system, that resource type is also included in the prompt:

```
zonecfg:zonename:fs>
```

For more information, including procedures that show how to use the various `zonecfg` components described in this chapter, see Chapter 17, "Planning and Configuring Non-Global Zones (Tasks)."

`zonecfg` Modes

The concept of a *scope* is used for the user interface. The scope can be either *global* or *resource specific*. The default scope is global.

In the global scope, the `add` subcommand and the `select` subcommand are used to select a specific resource. The scope then changes to that resource type.

- For the `add` subcommand, the `end` or `cancel` subcommands are used to complete the resource specification.
- For the `select` subcommand, the `end` or `cancel` subcommands are used to complete the resource modification.

The scope then reverts back to global.

Certain subcommands, such as `add`, `remove`, and `set`, have different semantics in each scope.

`zonecfg` Interactive Mode

In interactive mode, the following subcommands are supported. For detailed information about semantics and options used with the subcommands, see the `zonecfg`(1M) man page for options. For any subcommand that could result in destructive actions or loss of work, the system requests user confirmation before proceeding. You can use the `-F` (force) option to bypass this confirmation.

`help` Print general help, or display help about a given resource.

```
zonecfg:my-zone:capped-cpu> help
```

`create` Begin configuring an in-memory configuration for the specified new zone for one of these purposes:

- To apply the Sun default settings to a new configuration. This method is the default.
- With the `-t` *template* option, to create a configuration that is identical to the specified template. The zone name is changed from the template name to the new zone name.
- With the `-F` option, to overwrite an existing configuration.
- With the `-b` option, to create a blank configuration in which nothing is set.

`export` Print the configuration to standard output, or to the output file specified, in a form that can be used in a command file.

`add` — In the global scope, add the specified resource type to the configuration.

In the resource scope, add a property of the given name with the given value.

See "How to Configure the Zone" on page 238 and the zonecfg(1M) man page for more information.

`set` — Set a given property name to the given property value. Note that some properties, such as zonepath, are global, while others are resource specific. Thus, this command is applicable in both the global and resource scopes.

`select` — Applicable only in the global scope. Select the resource of the given type that matches the given property name-property value pair criteria for modification. The scope is changed to that resource type. You must specify a sufficient number of property name-value pairs for the resource to be uniquely identified.

`clear` — Clear the value for optional settings. Required settings cannot be cleared. However, some required settings can be changed by assigning a new value.

`remove` — In the global scope, remove the specified resource type. You must specify a sufficient number of property name-value pairs for the resource type to be uniquely identified. If no property name-value pairs are specified, all instances will be removed. If more than one exists, a confirmation is required unless the -F option is used.

In the resource scope, remove the specified property name-property value from the current resource.

`end` — Applicable only in the resource scope. End the resource specification.

The zonecfg command then verifies that the current resource is fully specified.

- If the resource is fully specified, it is added to the in-memory configuration and the scope will revert back to global.
- If the specification is incomplete, the system displays an error message that describes what needs to be done.

`cancel` — Applicable only in the resource scope. End the resource specification and reset the scope to global. Any partially specified resources are not retained.

`delete` — Destroy the specified configuration. Delete the configuration both from memory and from stable storage. You must use the -F (force) option with delete.

Caution – This action is instantaneous. No commit is required, and a deleted zone cannot be reverted.

`info` Display information about the current configuration or the global resource properties `zonepath`, `autoboot`, and `pool`. If a resource type is specified, display information only about resources of that type. In the resource scope, this subcommand applies only to the resource being added or modified.

`verify` Verify current configuration for correctness. Ensure that all resources have all of their required properties specified.

`commit` Commit current configuration from memory to stable storage. Until the in-memory configuration is committed, changes can be removed with the `revert` subcommand. A configuration must be committed to be used by `zoneadm`. This operation is attempted automatically when you complete a `zonecfg` session. Because only a correct configuration can be committed, the commit operation automatically does a verify.

`revert` Revert configuration back to the last committed state.

`exit` Exit the `zonecfg` session. You can use the `-F` (force) option with `exit`.

A `commit` is automatically attempted if needed. Note that an EOF character can also be used to exit the session.

`zonecfg` Command-File Mode

In command-file mode, input is taken from a file. The `export` subcommand described in "`zonecfg` Interactive Mode" on page 219 is used to produce this file. The configuration can be printed to standard output, or the `-f` option can be used to specify an output file.

Zone Configuration Data

Zone configuration data consists of two kinds of entities: resources and properties. Each resource has a type, and each resource can also have a set of one or more properties. The properties have names and values. The set of properties is dependent on the resource type.

The only required properties are `zonename` and `zonepath`.

Resource and Property Types

The resource and property types are described as follows:

`zonename` The name of the zone. The following rules apply to zone names:

- Each zone must have a unique name.

- A zone name is case-sensitive.
- A zone name must begin with an alphanumeric character.

  The name can contain alphanumeric characters, underbars (_), hyphens (-), and periods (.).
- The name cannot be longer than 64 characters.
- The name `global` and all names beginning with `SUNW` are reserved and cannot be used.

`zonepath`

The `zonepath` property is the path to the zone root. Each zone has a path to its root directory that is relative to the global zone's root directory. At installation time, the global zone directory is required to have restricted visibility. It must be owned by `root` with the mode `700`.

The non-global zone's root path is one level lower. The zone's root directory has the same ownership and permissions as the root directory (`/`) in the global zone. The zone directory must be owned by `root` with the mode `755`. These directories are created automatically with the correct permissions, and do not need to be verified by the zone administrator. This hierarchy ensures that unprivileged users in the global zone are prevented from traversing a non-global zone's file system.

The zone must reside on a ZFS dataset. The ZFS dataset will be created automatically when the zone is installed or attached. If a ZFS dataset cannot be created, the zone will not install or attach. The parent directory of the zone path must also be a dataset.

Path	Description
`/zones/my-zone`	`zonecfg zonepath`
`/zones/my-zone/root`	Root of the zone
`/zones/my-zone/root/dev`	Devices created for the zone

See “Traversing File Systems” on page 315 for a further discussion of this issue.

Note – You can move a zone to another location on the same system by specifying a new, full `zonepath` with the `move` subcommand of `zoneadm`. See “Moving a Non-Global Zone” on page 293 for instructions.

`autoboot` — If this property is set to true, the zone is automatically booted when the global zone is booted. Note that if the zones service `svc:/system/zones:default` is disabled, the zone will not automatically boot, regardless of the setting of this property. You can enable the zones service with the `svcadm` command described in the `svcadm`(1M) man page:

```
global# svcadm enable zones
```

See "Zones Packaging Overview" on page 301 for information on this setting during `pkg image-update`.

`bootargs` — This property is used to set a boot argument for the zone. The boot argument is applied unless overridden by the `reboot`, `zoneadm boot`, or `zoneadm reboot` commands. See "Zone Boot Arguments" on page 258.

`pool` — This property is used to associate the zone with a resource pool on the system. Multiple zones can share the resources of one pool. Also see "`dedicated-cpu` Resource" on page 207.

`limitpriv` — This property is used to specify a privilege mask other than the default. See "Privileges in a Non-Global Zone" on page 324.

Privileges are added by specifying the privilege name, with or without the leading `priv_`. Privileges are excluded by preceding the name with a dash (`-`) or an exclamation mark (`!`). The privilege values are separated by commas and placed within quotation marks (").

As described in `priv_str_to_set`(3C), the special privilege sets of `none`, `all`, and `basic` expand to their normal definitions. Because zone configuration takes place from the global zone, the special privilege set `zone` cannot be used. Because a common use is to alter the default privilege set by adding or removing certain privileges, the special set `default` maps to the default, set of privileges. When `default` appears at the beginning of the `limitpriv` property, it expands to the default set.

The following entry adds the ability to use DTrace programs that only require the `dtrace_proc` and `dtrace_user` privileges in the zone:

```
global# zonecfg -z userzone
zonecfg:userzone> set limitpriv="default,dtrace_proc,dtrace_user"
```

If the zone's privilege set contains a disallowed privilege, is missing a required privilege, or includes an unknown privilege, an attempt to verify, ready, or boot the zone will fail with an error message.

`scheduling-class` — This property sets the scheduling class for the zone. See "Scheduling Class" on page 208 for additional information and tips.

`ip-type`

This property is required to be set only if the zone is an exclusive-IP zone. See "Exclusive-IP Non-Global Zones" on page 211 and "How to Configure the Zone" on page 238.

`dedicated-cpu`

This resource dedicates a subset of the system's processors to the zone while it is running. The `dedicated-cpu` resource provides limits for `ncpus` and, optionally, `importance`. For more information, see "`dedicated-cpu` Resource" on page 207.

`capped-cpu`

This resource sets a limit on the amount of CPU resources that can be consumed by the zone while it is running. The `capped-cpu` resource provides a limit for `ncpus`. For more information, see "`capped-cpu` Resource" on page 208.

`capped-memory`

This resource groups the properties used when capping memory for the zone. The `capped-memory` resource provides limits for `physical`, `swap`, and `locked` memory. At least one of these properties must be specified.

`net`

The network interface resource is the interface name. Each zone can have network interfaces that should be set up when the zone transitions from the installed state to the ready state.

`dataset`

Adding a ZFS dataset resource enables the delegation of storage administration to a non-global zone. The zone administrator can create and destroy file systems within that dataset, and modify properties of the dataset. The zone administrator cannot affect datasets that have not been added to the zone or exceed any top level quotas set on the dataset assigned to the zone. After a dataset is delegated to a non-global zone, the `zoned` property is automatically set. A `zoned` file system cannot be mounted in the global zone because the zone administrator might have to set the mount point to an unacceptable value.

ZFS datasets can be added to a zone in the following ways.

- As an lofs mounted file system, when the goal is solely to share space with the global zone
- As a delegated dataset

See Chapter 10, "Oracle Solaris ZFS Advanced Topics," in *Oracle Solaris ZFS Administration Guide* and "File Systems and Non-Global Zones" on page 310.

Also see Chapter 26, "Troubleshooting Miscellaneous Oracle Solaris Zones Problems," for information on dataset issues.

`fs`

Each zone can have various file systems that are mounted when the zone transitions from the installed state to the ready state. The file system resource specifies the path to the file system mount point. For more information about the use of file systems in zones, see "File Systems and Non-Global Zones" on page 310.

`fs-allowed`

Setting this property gives the zone administrator the ability to mount any file system of that type, either created by the zone administrator or imported by using NFS, and administer that file system. File system mounting permissions within a running zone are also restricted by the `fs-allowed` property. By default, only mounts of `hsfs` file systems and network file systems, such as NFS, are allowed within a zone.

The property can be used with a block device or ZVOL device delegated into the zone as well.

The `fs-allowed` property accepts a comma-separated list of additional file systems that can be mounted from within the zone, for example, `ufs,pcfs`.

```
zonecfg:my-zone> set fs-allowed=ufs,pcfs
```

This property does not affect zone mounts administrated by the global zone through the `add fs` or `add dataset` properties.

For security considerations, see "File Systems and Non-Global Zones" on page 310 and "Device Use in Non-Global Zones" on page 320.

`device`

The device resource is the device matching specifier. Each zone can have devices that should be configured when the zone transitions from the installed state to the ready state.

`rctl`

The `rctl` resource is used for zone-wide resource controls. The controls are enabled when the zone transitions from the installed state to the ready state.

See "Setting Zone-Wide Resource Controls" on page 214 for more information.

Note – To configure zone-wide controls using the `set` *global_property_name* subcommand of `zonefig` instead of the `rctl` resource, see "How to Configure the Zone" on page 238.

`hostid`

A `hostid` that is different from the `hostid` of the global zone can be set.

`attr`

This generic attribute can be used for user comments or by other subsystems. The `name` property of an `attr` must begin with an alphanumeric character. The `name` property can contain alphanumeric characters, hyphens (`-`), and periods (`.`). Attribute names beginning with `zone.` are reserved for use by the system.

Resource Type Properties

Resources also have properties to configure. The following properties are associated with the resource types shown.

`admin` Define the user name and the authorizations for that user for a given zone.

```
zonecfg:my-zone> add admin
zonecfg:my-zone:admin> set user=zadmin
zonecfg:my-zone:admin> set auths=login,manage
zonecfg:my-zone:admin> end
```

The following values can be used for the `auths` property:

- `login` (`solaris.zone.login`)
- `manage` (`solaris.zone.manage`)
- `clone` (`solaris.zone.clonefrom`)

Note that these `auths` do not enable you to create a zone. This capability is included in the Zone Security profile.

`dedicated-cpu` `ncpus`, `importance`

Specify the number of CPUs and, optionally, the relative importance of the pool. The following example specifies a CPU range for use by the zone `my-zone`. `importance` is also set.

```
zonecfg:my-zone> add dedicated-cpu
zonecfg:my-zone:dedicated-cpu> set ncpus=1-3
zonecfg:my-zone:dedicated-cpu> set importance=2
zonecfg:my-zone:dedicated-cpu> end
```

`capped-cpu` `ncpus`

Specify the number of CPUs. The following example specifies a CPU cap of 3.5 CPUs for the zone `my-zone`.

```
zonecfg:my-zone> add capped-cpu
zonecfg:my-zone:capped-cpu> set ncpus=3.5
zonecfg:my-zone:capped-cpu> end
```

`capped-memory` `physical`, `swap`, `locked`Specify the memory limits for the zone `my-zone`. Each limit is optional, but at least one must be set.

```
zonecfg:my-zone> add capped-memory
zonecfg:my-zone:capped-memory> set physical=50m
zonecfg:my-zone:capped-memory> set swap=100m
zonecfg:my-zone:capped-memory> set locked=30m
zonecfg:my-zone:capped-memory> end
```

`fs` `dir`, `special`, `raw`, `type`, `options`

The `fs` resource parameters supply the values that determine how and where to mount file systems. The `fs` parameters are defined as follows:

`dir` Specifies the mount point for the file system

`special` Specifies the block special device name or directory from the global zone to mount

`raw` Specifies the raw device on which to run `fsck` before mounting the file system (not applicable to ZFS)

`type` Specifies the file system type

`options` Specifies mount options similar to those found with the `mount` command

The lines in the following example specify that the dataset named `pool1/fs1` in the global zone is to be mounted as `/shared/fs1` in a zone being configured. The file system type to use is ZFS.

```
zonecfg:my-zone> add fs
zonecfg:my-zone:fs> set dir=/shared/fs1
zonecfg:my-zone:fs> set special=pool1/fs1
zonecfg:my-zone:fs> set type=zfs
zonecfg:my-zone:fs> end
```

For more information on parameters, see "The `-o nosuid` Option" on page 310, "Security Restrictions and File System Behavior" on page 312, and the `fsck`(1M) and `mount`(1M) man pages. Also note that section 1M man pages are available for mount options that are unique to a specific file system. The names of these man pages have the form `mount_`*filesystem*.

`dataset` `name`

The lines in the following example specify that the dataset *sales* is to be visible and mounted in the non-global zone and no longer visible in the global zone.

```
zonecfg:my-zone> add dataset
zonecfg:my-zone> set name=tank/sales
zonecfg:my-zone> end
```

`net` `address`, `allowed-addressphysical`, `defrouter`

Note – For a shared-IP zone, both the IP address and the device are specified. Optionally, the default router can be set. For an exclusive-IP zone, only the physical interface must be specified.

- The `allowed-address` property limits the set of configurable IP addresses that can be used by an exclusive-IP zone.
- The `defrouter` property can be used to set a default route when the non-global zone and the global zone reside on separate networks.
- Any zone that has the `defrouter` property set must be on a subnet that is not configured for the global zone.
- Traffic from a zone with a default router will go out to the router before coming back to the destination zone.

When shared-IP zones exist on different subnets, do not configure a data-link in the global zone.

For an exclusive-IP zone, the physical property can be a VNIC.

In the following example for a shared-IP zone, the IP address `192.168.0.1` is added to the zone. An `hme0` card is used for the physical interface. To determine which physical interface to use, type `ifconfig -a` on your system. Each line of the output, other than loopback driver lines, begins with the name of a card installed on your system. Lines that contain `LOOPBACK` in the descriptions do not apply to cards. The default route is set to `10.0.0.1` for the zone.

```
zonecfg:my-zone> add net
zonecfg:my-zone:net> set physical=hme0
zonecfg:my-zone:net> set address=192.168.0.1
zonecfg:my-zone:net> set defrouter=10.0.0.1
zonecfg:my-zone:net> end
```

In the following example for an exclusive-IP zone, a `bge32001` link is used for the physical interface, which is a VLAN on `bge1`. To determine which data-links are available, use the command `dladm show-link`. The `allowed-address` property constrains which IP addresses zone can use. The `defrouter` property is used to set a default route. Note that `ip-type=exclusive` must also be specified.

```
zonecfg:my-zone> set ip-type=exclusive
zonecfg:my-zone> add net
zonecfg:myzone:net> set allowed-address=11.1.1.32/24
zonecfg:my-zone:net> set physical=vnic0
zonecfg:myzone:net> set defrouter=11.1.1.1
zonecfg:my-zone:net> end
```

The `physical` property can be a VNIC, as described in Part IV, " Network Virtualization and Resource Management," in *System Administration Guide: Network Interfaces and Network Virtualization*.

Note – The Oracle Solaris operating system supports all Ethernet-type interfaces, and their data-links can be administered with the `dladm` command.

`device` `match`

In the following example, a `/dev/pts` device is included in a zone.

```
zonecfg:my-zone> add device
zonecfg:my-zone:device> set match=/dev/pts*
zonecfg:my-zone:device> end
```

Caution – Before adding devices, see "Device Use in Non-Global Zones" on page 320, "Running Applications in Non-Global Zones" on page 322, and "Privileges in a Non-Global Zone" on page 324 for restrictions and security concerns.

`rctl` `name`, `value`

The following zone-wide resource controls are available.

- `zone.cpu-cap`
- `zone.cpu-shares` (preferred: `cpu-shares`)
- `zone.max-locked-memory`
- `zone.max-lofi`
- `zone.max-lwps` (preferred: `max-lwps`)
- `zone.max-msg-ids` (preferred: `max-msg-ids`)
- `zone.max-processes`(preferred: `max-processes`
- `zone.max-sem-ids` (preferred: `max-sem-ids`)
- `zone.max-shm-ids` (preferred: `max-shm-ids`)
- `zone.max-shm-memory` (preferred: `max-shm-memory`)
- `zone.max-swap`

Note that the preferred, simpler method for setting a zone-wide resource control is to use the property name instead of the `rctl` resource, as shown in "How to Configure the Zone" on page 238. If zone-wide resource control entries in a zone are configured using `add rctl`, the format is different than resource control entries in the `project` database. In a zone configuration, the `rctl` resource type consists of three name/value pairs. The names are `priv`, `limit`, and `action`. Each of the names takes a simple value.

```
zonecfg:my-zone> add rctl
zonecfg:my-zone:rctl> set name=zone.cpu-shares
zonecfg:my-zone:rctl> add value (priv=privileged,limit=10,action=none)
zonecfg:my-zone:rctl> end

zonecfg:my-zone> add rctl
zonecfg:my-zone:rctl> set name=zone.max-lwps
zonecfg:my-zone:rctl> add value (priv=privileged,limit=100,action=deny)
zonecfg:my-zone:rctl> end
```

For general information about resource controls and attributes, see Chapter 6, "Resource Controls (Overview)," and "Resource Controls Used in Non-Global Zones" on page 322.

`attr` `name`, `type`, `value`

In the following example, a comment about a zone is added.

```
zonecfg:my-zone> add attr
zonecfg:my-zone:attr> set name=comment
zonecfg:my-zone:attr> set type=string
zonecfg:my-zone:attr> set value="Production zone"
zonecfg:my-zone:attr> end
```

You can use the `export` subcommand to print a zone configuration to standard output. The configuration is saved in a form that can be used in a command file.

Tecla Command-Line Editing Library

The Tecla command-line editing library is included for use with the `zonecfg` command. The library provides a mechanism for command-line history and editing support.

The Tecla command-line editing library is documented in the following man pages:

- `enhance`(1)
- `libtecla`(3LIB)
- `ef_expand_file`(3TECLA)
- `gl_get_line`(3TECLA)
- `gl_io_mode`(3TECLA)
- `pca_lookup_file`(3TECLA)
- `tecla`(5)

CHAPTER 17

Planning and Configuring Non-Global Zones (Tasks)

This chapter describes what you need to do before you can configure a zone on your system. This chapter also describes how to configure a zone, modify a zone configuration, and delete a zone configuration from your system.

For an introduction to the zone configuration process, see Chapter 16, "Non-Global Zone Configuration (Overview)."

For information about `solaris10` branded zone configuration, see Part III, "Oracle Solaris 10 Zones."

Planning and Configuring a Non-Global Zone (Task Map)

Before you set up your system to use zones, you must first collect information and make decisions about how to configure the zones. The following task map summarizes how to plan and configure a zone.

Task	Description	For Instructions
Plan your zone strategy.	■ Evaluate the applications running on your system to determine which applications you want to run in a zone. ■ Assess the availability of disk space to hold the files that are unique in the zone. ■ If you are also using resource management features, determine how to align the zone with the resource management boundaries. ■ If you are using resource pools, configure the pools if necessary.	Refer to historical usage. Also see "Disk Space Requirements" on page 234 and "Resource Pools Used in Zones" on page 137.
Determine the name for the zone.	Decide what to call the zone based on the naming conventions.	See "Zone Configuration Data" on page 221 and "Zone Host Name" on page 235.
Determine the zone path (required).	Each zone has a path to its root directory that is relative to the global zone's root directory.	See "Zone Configuration Data" on page 221.
Evaluate the need for CPU restriction if you are not configuring resource pools. Note that with specification in `zonecfg`, pool settings propagate during migrations.	Review your application requirements.	See "`dedicated-cpu` Resource" on page 207.
Evaluate the need for memory allocation if you plan to cap memory for the zone by using `rcapd` from the global zone.	Review your application requirements.	See Chapter 10, "Physical Memory Control Using the Resource Capping Daemon (Overview)," Chapter 11, "Administering the Resource Capping Daemon (Tasks)," and "Physical Memory Control and the `capped-memory` Resource" on page 209.

Task	Description	For Instructions
Make the FSS the default scheduler on the system.	Give each zone CPU shares to control the zone's entitlement to CPU resources. The FSS guarantees a fair dispersion of CPU resources among zones that is based on allocated shares.	Chapter 8, "Fair Share Scheduler (Overview)," "Scheduling Class" on page 208.
Determine whether the zone will be a shared-IP zone or an exclusive-IP zone.	For a shared-IP zone, which is the default, obtain or configure IP addresses for the zone. Depending on your configuration, you must obtain at least one IP address for each non-global zone that you want to have network access. For an exclusive-IP zone, determine the data-link that will be assigned to the zone. The zone requires exclusive access to one or more network interfaces. The interface could be a separate LAN such as `bge1`, or a separate VLAN such as `bge2000`.	See "Determine the Zone Host Name and the Network Requirements" on page 235, "How to Configure the Zone" on page 238, and *System Administration Guide: IP Services*.
Determine which file systems you want to mount in the zone.	Review your application requirements.	See "File Systems Mounted in Zones" on page 213 for more information.
Determine which network interfaces should be made available in the zone.	Review your application requirements.	See "Shared-IP Network Interfaces" on page 317 for more information.
Determine whether you must alter the default set of non-global zone permissions.	Check the set of privileges: default, privileges that can be added and removed, and privileges that cannot be used at this time.	See "Privileges in a Non-Global Zone" on page 324.
Determine which devices should be configured in each zone.	Review your application requirements.	Refer to the documentation for your application.
Configure the zone.	Use `zonecfg` to create a configuration for the zone.	See "Configuring, Verifying, and Committing a Zone" on page 238.
Verify and commit the configured zone.	Determine whether the resources and properties specified are valid on a hypothetical system.	See "Configuring, Verifying, and Committing a Zone" on page 238.

Evaluating the Current System Setup

Zones can be used on any machine that runs the Oracle Solaris 10 or later release. The following primary machine considerations are associated with the use of zones.

- The performance requirements of the applications running within each zone.
- The availability of disk space to hold the files that are unique within each zone.

Disk Space Requirements

There are no limits on how much disk space can be consumed by a zone. The global administrator or a user with appropriate authorizations is responsible for space restriction. The global administrator must ensure that local storage is sufficient to hold a non-global zone's root file system. Even a small uniprocessor system can support a number of zones running simultaneously.

The nature of the packages installed in the non-global zone affects the space requirements of the zones. The number of packages is also a factor.

The whole root zone model provides the maximum configurability. All of the required and any selected optional Oracle Solaris packages are installed into the private file systems of the zone. The advantages of this model include the capability for global administrators to customize the file system layout of their zones. This would be done, for example, to add arbitrary unbundled or third-party packages.

The disk requirements for this model are determined by the disk space used by the packages currently installed in the global zone and the installed software.

A zone requires a minimum of 150 megabytes of free disk space per zone. However, the free disk space needed is generally from 500 megabytes to 1 gigabyte when the global zone has been installed with all of the standard Oracle Solaris packages. That figure can increase if more software is added.

An additional 40 megabytes of RAM per zone are suggested, but not required on a machine with sufficient swap space.

Restricting Zone Size

You can use ZFS dataset quotas with zones that have `zonepaths` backed by ZFS datasets to restrict zone size. Administrators that can access `zonepath` datasets can modify the datasets' `quota`, `userquota`, `groupquota`, and `refquota` properties to control the maximum amount of disk space that each zone can consume. These properties are described in the `zfs`(1M) man page.

Administrators can also create ZFS volumes with fixed sizes and install zones in the volume's datasets. The volumes will limit the sizes of the zones installed within them.

Determine the Zone Host Name and the Network Requirements

You must determine the host name for the zone. Then, for a shared IP zone that will have network connectivity, you must do one of the following:

- Assign an IPv4 address for the zone
- Manually configure and assign an IPv6 address for the zone

Inside an exclusive-IP zone, you configure addresses as you do for the global zone.

For more information on IP types, see "Zone Network Interfaces" on page 210

Zone Host Name

The host name you select for the zone must be defined either in the `hosts` database or in the `/etc/inet/hosts` database, as specified by the `/etc/nsswitch.conf` file in the global zone. The network databases are files that provide network configuration information. The `nsswitch.conf` file specifies which naming service to use.

If you use local files for the naming service, the `hosts` database is maintained in the `/etc/inet/hosts` file. The host names for zone network interfaces are resolved from the local `hosts` database in `/etc/inet/hosts`. Alternatively, the IP address itself can be specified directly when configuring a zone so that no host name resolution is required.

For more information, see "TCP/IP Configuration Files" in *System Administration Guide: IP Services* and "Network Databases and the nsswitch.conf File" in *System Administration Guide: IP Services*.

Shared-IP Zone Network Address

Each shared-IP zone that requires network connectivity has one or more unique IP addresses. Both IPv4 and IPv6 addresses are supported.

IPv4 Zone Network Address

If you are using IPv4, obtain an address and assign the address to the zone.

A prefix length can also be specified with the IP address. The format of this prefix is *address/prefix-length*, for example, `192.168.1.1/24`. Thus, the address to use is `192.168.1.1` and the netmask to use is `255.255.255.0`, or the mask where the first 24 bits are 1-bits.

IPv6 Zone Network Address

If you are using IPv6, you must manually configure the address. Typically, at least the following two types of addresses must be configured:

Link-local address
: A link-local address is of the form `fe80::`*64-bit interface ID*`/10`. The `/10` indicates a prefix length of 10 bits.

Address formed from a global prefix configured on the subnet
: A global unicast address is based off a 64–bit prefix that the administrator configures for each subnet, and a 64-bit interface ID. The prefix can also be obtained by running the `ifconfig` command with the `-a6` option on any system on the same subnet that has been configured to use IPv6.

  The 64–bit interface ID is typically derived from a system's MAC address. For zones use, an alternate address that is unique can be derived from the global zone's IPv4 address as follows:

```
16 bits of zero:upper 16 bits of IPv4 address:lower 16 bits of IPv4 address:a
zone-unique number
```

  For example, if the global zone's IPv4 address is 192.168.200.10, a suitable link-local address for a non-global zone using a zone-unique number of 1 is `fe80::c0a8:c80a:1/10`. If the global prefix in use on that subnet is `2001:0db8:aabb:ccdd/64`, a unique global unicast address for the same non-global zone is `2001:0db8:aabb:ccdd::c0a8:c80a:1/64`. Note that you must specify a prefix length when configuring an IPv6 address.

For more information about link-local and global unicast addresses, see the `inet6(7P)` ma page.

Exclusive-IP Zone Network Address

Inside an exclusive-IP zone, configure addresses as you do for the global zone. Note that DHCP and IPv6 stateless address autoconfiguration can be used to configure addresses.

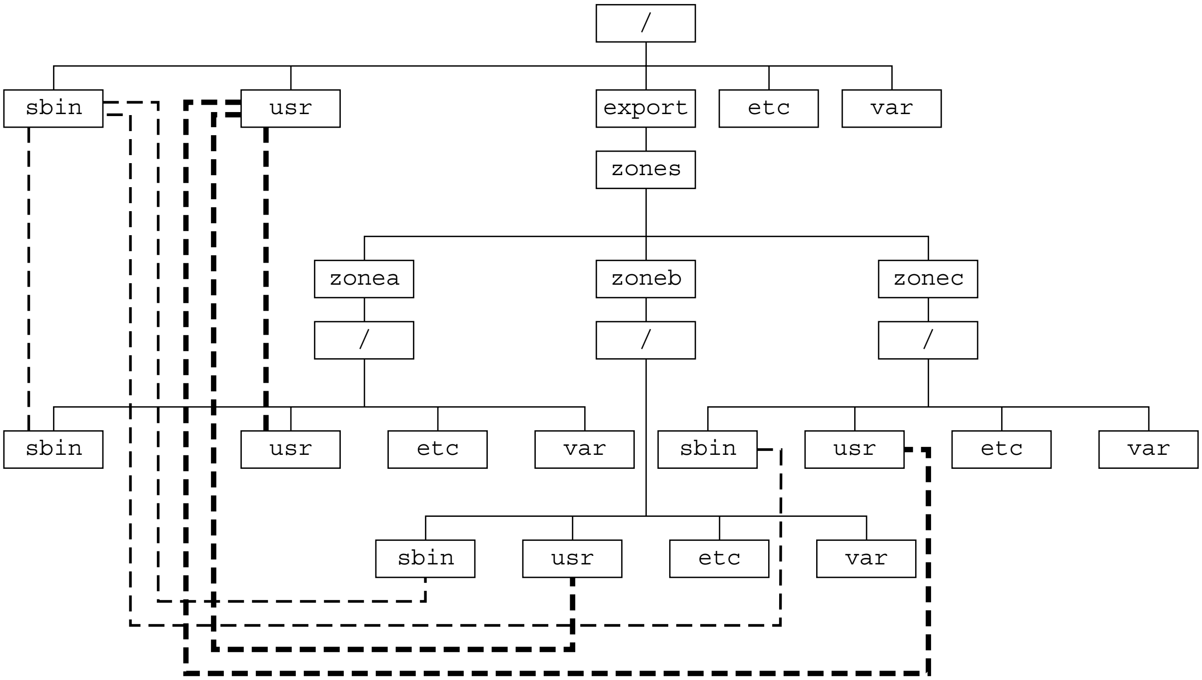
See `sysidcfg(4)` for more information.

File System Configuration

You can specify a number of mounts to be performed when the virtual platform is set up. File systems that are loopback-mounted into a zone by using the loopback virtual file system (LOFS) file system should be mounted with the `nodevices` option. For information on the `nodevices` option, see “File Systems and Non-Global Zones” on page 310.

LOFS lets you create a new virtual file system so that you can access files by using an alternative path name. In a non-global zone, a loopback mount makes the file system hierarchy look as though it is duplicated under the zone's root. In the zone, all files will be accessible with a path name that starts from the zone's root. LOFS mounting preserves the file system name space.

FIGURE 17–1 Loopback-Mounted File Systems



See the `lofs`(7S) man page for more information.

Creating, Revising, and Deleting Non-Global Zone Configurations (Task Map)

Task	Description	For Instructions
Configure a non-global zone.	Use the `zonecfg` command to create a zone, verify the configuration, and commit the configuration. You can also use a script to configure and boot multiple zones on your system. You can use the `zonecfg` command to display the configuration of a non-global zone.	"Configuring, Verifying, and Committing a Zone" on page 238, "Script to Configure Multiple Zones" on page 243

Task	Description	For Instructions
Modify a zone configuration.	Use these procedures to modify a resource type in a zone configuration, modify a property type such as the name of a zone, or add a dedicated device to a zone.	"Using the zonecfg Command to Modify a Zone Configuration" on page 245
Revert a zone configuration or delete a zone configuration.	Use the zonecfg command to undo a resource setting made to a zone configuration or to delete a zone configuration.	"Using the zonecfg Command to Revert or Remove a Zone Configuration" on page 249
Delete a zone configuration.	Use the zonecfg command with the delete subcommand to delete a zone configuration from the system.	"How to Delete a Zone Configuration" on page 250

Configuring, Verifying, and Committing a Zone

The zonecfg command described in the zonecfg(1M) man page is used to perform the following actions.

- Create the zone configuration
- Verify that all required information is present
- Commit the non-global zone configuration

The zonecfg command can also be used to persistently specify the resource management settings for the global zone.

While configuring a zone with the zonecfg utility, you can use the revert subcommand to undo the setting for a resource. See "How to Revert a Zone Configuration" on page 249.

A script to configure multiple zones on your system is provided in "Script to Configure Multiple Zones" on page 243.

To display a non-global zone's configuration, see "How to Display the Configuration of a Non-Global Zone" on page 245.

▼ How to Configure the Zone

Note that the only required elements to create a non-global zone are the zonename and zonepath properties. Other resources and properties are optional. Some optional resources also require choices between alternatives, such as the decision to use either the dedicated-cpu resource or the capped-cpu resource. See "Zone Configuration Data" on page 221 for information on available zonecfg properties and resources.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have the correct rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Set up a zone configuration with the zone name you have chosen.**

   The name `my-zone` is used in this example procedure.

   ```
   global# zonecfg -z my-zone
   ```

   If this is the first time you have configured this zone, you will see the following system message:

   ```
   my-zone: No such zone configured
   Use 'create' to begin configuring a new zone.
   ```

3. **Create the new zone configuration.**

   This procedure uses the Sun default settings.

   ```
   zonecfg:my-zone> create
   ```

4. **Set the zone path, `/zones/my-zone` in this procedure.**

   ```
   zonecfg:my-zone> set zonepath=/zones/my-zone
   ```

   The zone must reside on a ZFS dataset. The ZFS dataset will be created automatically when the zone is installed or attached. If a ZFS dataset cannot be created, the zone will not install or attach. Note that the parent directory of the zone path must also be a dataset.

5. **Set the autoboot value.**

   If set to `true`, the zone is automatically booted when the global zone is booted. Note that for the zones to autoboot, the zones service `svc:/system/zones:default` must also be enabled. The default value is `false`.

   ```
   zonecfg:my-zone> set autoboot=true
   ```

6. **Set persistent boot arguments for a zone.**

   ```
   zonecfg:my-zone> set bootargs="-m verbose"
   ```

7. **Dedicate one CPU to this zone.**

   ```
   zonecfg:my-zone> add dedicated-cpu
   ```

   a. **Set the number of CPUs.**

      ```
      zonecfg:my-zone:dedicated-cpu> set ncpus=1-2
      ```

   b. **(Optional) Set the importance.**

      ```
      zonecfg:my-zone:dedicated-cpu> set importance=10
      ```

The default is `1`.

c. **End the specification.**

```
zonecfg:my-zone:dedicated-cpu> end
```

8 **Revise the default set of privileges.**

```
zonecfg:my-zone> set limitpriv="default,sys_time"
```

This line adds the ability to set the system clock to the default set of privileges.

9 **Set the scheduling class to `FSS`.**

```
zonecfg:my-zone> set scheduling-class=FSS
```

10 **Add a memory cap.**

```
zonecfg:my-zone> add capped-memory
```

a. **Set the memory cap.**

```
zonecfg:my-zone:capped-memory> set physical=50m
```

b. **Set the swap memory cap.**

```
zonecfg:my-zone:capped-memory> set swap=100m
```

c. **Set the locked memory cap.**

```
zonecfg:my-zone:capped-memory> set locked=30m
```

d. **End the memory cap specification.**

```
zonecfg:my-zone:capped-memory> end
```

11 **Add a file system.**

```
zonecfg:my-zone> add fs
```

a. **Set the mount point for the file system, `/usr/local` in this procedure.**

```
zonecfg:my-zone:fs> set dir=/usr/local
```

b. **Specify that `/opt/local` in the global zone is to be mounted as `/usr/local` in the zone being configured.**

```
zonecfg:my-zone:fs> set special=/opt/local
```

In the non-global zone, the `/usr/local` file system will be readable and writable.

c. **Specify the file system type, `lofs` in this procedure.**

```
zonecfg:my-zone:fs> set type=lofs
```

The type indicates how the kernel interacts with the file system.

d. **End the file system specification.**

```
zonecfg:my-zone:fs> end
```

This step can be performed more than once to add more than one file system.

12 **Set the `hostid` if necessary.**

```
zonecfg:my-zone> set hostid=80f0c086
```

13 **Add a ZFS dataset named *sales* in the storage pool *tank***

```
zonecfg:my-zone> add dataset
```

a. **Specify the path to the ZFS dataset *sales*.**

```
zonecfg:my-zone> set name=tank/sales
```

b. **End the dataset specification.**

```
zonecfg:my-zone> end
```

The zone administrator can create and destroy file systems within the dataset, and modify properties of the dataset.

14 **(Optional) If you are creating an exclusive-IP zone, set the `ip-type`.**

```
zonecfg:my-zone> set ip-type=exclusive
```

Note – Only the physical device type will be specified in the `add net` step. The `physical` property can be a VNIC, as described in Part IV, "Network Virtualization and Resource Management," in *System Administration Guide: Network Interfaces and Network Virtualization*.

---

15 **Add a network interface.**

```
zonecfg:my-zone> add net
```

a. **(shared-IP only) Set the IP address for the network interface, `192.168.0.1` in this procedure.**

```
zonecfg:my-zone:net> set address=192.168.0.1
```

b. **Set the physical device type for the network interface, the `hme` device in this procedure.**

```
zonecfg:my-zone:net> set physical=hme0
```

c. **(shared-IP only) Set the default router for the network interface, in this procedure.**

```
zonecfg:my-zone:net> set defrouter=10.0.0.1
```

Setting the `defrouter` property for the `net` resource is optional.

d. **End the specification.**

```
zonecfg:my-zone:net> end
```

This step can be performed more than once to add more than one network interface.

16 Add a device.

```
zonecfg:my-zone> add device
```

a. Set the device match, `/dev/sound/*` in this procedure.

```
zonecfg:my-zone:device> set match=/dev/sound/*
```

b. End the device specification.

```
zonecfg:my-zone:device> end
```

This step can be performed more than once to add more than one device.

17 Add a zone-wide resource control by using the property name.

```
zonecfg:my-zone> set max-sem-ids=10485200
```

This step can be performed more than once to add more than one resource control.

18 Add a comment by using the `attr` resource type.

```
zonecfg:my-zone> add attr
```

a. Set the name to `comment`.

```
zonecfg:my-zone:attr> set name=comment
```

b. Set the type to `string`.

```
zonecfg:my-zone:attr> set type=string
```

c. Set the value to a comment that describes the zone.

```
zonecfg:my-zone:attr> set value="This is my work zone."
```

d. End the `attr` resource type specification.

```
zonecfg:my-zone:attr> end
```

19 Verify the zone configuration for the zone.

```
zonecfg:my-zone> verify
```

20 Commit the zone configuration for the zone.

```
zonecfg:my-zone> commit
```

21 Exit the `zonecfg` command.

```
zonecfg:my-zone> exit
```

Note that even if you did not explicitly type `commit` at the prompt, a `commit` is automatically attempted when you type `exit` or an EOF occurs.

More Information Using Multiple Subcommands From the Command Line

---

Tip – The `zonecfg` command also supports multiple subcommands, quoted and separated by semicolons, from the same shell invocation.

```
global# zonecfg -z my-zone "create ; set zonepath=/zones/my-zone"
```

Where to Go From Here

See “Installing and Booting Zones” on page 264 to install your committed zone configuration.

Script to Configure Multiple Zones

You can use this script to configure and boot multiple zones on your system. The script takes the following parameters:

- The number of zones to be created
- The *zonename* prefix
- The directory to use as the base directory

You must be the global administrator with superuser privileges in the global zone or a user with the correct rights profile to execute the script.

```
#!/bin/ksh
#
# Copyright 2006-2010 Oracle Corporation. All rights reserved.
# Use is subject to license terms.
#
#ident    "%Z%%M%    %I%    %E% SMI"

if [[ -z "$1" || -z "$2" || -z "$3" ]]; then
       echo "usage: $0 <#-of-zones> <zonename-prefix> <basedir>"
       exit 2
fi

if [[ ! -d $3 ]]; then
      echo "$3 is not a directory"
       exit 1
fi

nprocs=‘psrinfo | wc -l‘
nzones=$1
prefix=$2
dir=$3

ip_addrs_per_if=‘ndd /dev/ip ip_addrs_per_if‘
if [ $ip_addrs_per_if -lt $nzones ]; then
       echo "ndd parameter ip_addrs_per_if is too low ($ip_addrs_per_if)"
```

```
     echo "set it higher with 'ndd -set /dev/ip ip_addrs_per_if <num>"
     exit 1
fi

i=1
while [ $i -le $nzones ]; do
    zoneadm -z $prefix$i list > /dev/null 2>&1
    if [ $? != 0 ]; then
        echo configuring $prefix$i
        F=$dir/$prefix$i.config
        rm -f $F
        echo "create" > $F
        echo "set zonepath=$dir/$prefix$i" >> $F
        zonecfg -z $prefix$i -f $dir/$prefix$i.config 2>&1 | \
            sed 's/^/    /g'
    else
        echo "skipping $prefix$i, already configured"
    fi
    i=`expr $i + 1`
done

i=1
while [ $i -le $nzones ]; do
    j=1
    while [ $j -le $nprocs ]; do
        if [ $i -le $nzones ]; then
            if [ `zoneadm -z $prefix$i list -p | \
                cut -d':' -f 3` != "configured" ]; then
                echo "skipping $prefix$i, already installed"
            else
                echo installing $prefix$i
                mkdir -pm 0700 $dir/$prefix$i
                chmod 700 $dir/$prefix$i
                zoneadm -z $prefix$i install > /dev/null 2>&1 &
                sleep 1    # spread things out just a tad
            fi
        fi
        i=`expr $i + 1`
        j=`expr $j + 1`
    done
    wait
done

i=1
while [ $i -le $nzones ]; do
    echo setting up sysid for $prefix$i
    cfg=$dir/$prefix$i/root/etc/sysidcfg
    rm -f $cfg
    echo "network_interface=NONE {hostname=$prefix$i}" > $cfg
    echo "system_locale=C" >> $cfg
    echo "terminal=xterms" >> $cfg
    echo "security_policy=NONE" >> $cfg
    echo "name_service=NONE" >> $cfg
    echo "timezone=US/Pacific" >> $cfg
    echo "root_password=Qexr7Y/wzkSbc" >> $cfg  # 'l1a'
    i=`expr $i + 1`
done

i=1
```

```
para=‘expr $nprocs \* 2‘
while [ $i -le $nzones ]; do
    date
    j=1
    while [ $j -le $para ]; do
        if [ $i -le $nzones ]; then
            echo booting $prefix$i
            zoneadm -z $prefix$i boot &
        fi
        j=‘expr $j + 1‘
        i=‘expr $i + 1‘
    done
    wait
done
```

▼ How to Display the Configuration of a Non-Global Zone

You must be the global administrator in the global zone or a user with the correct rights profile to perform this procedure.

1. **Be superuser, or have the correct rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Display the configuration of a zone.**

   ```
   global# zonecfg -z zonename info
   ```

Using the zonecfg Command to Modify a Zone Configuration

You can also use the zonecfg command to do the following:

- Modify a resource type in a zone configuration
- Clear a property value in a zone configuration
- Add a dedicated device to a zone

▼ How to Modify a Resource Type in a Zone Configuration

You can select a resource type and modify the specification for that resource.

You must be the global administrator in the global zone or a user with the correct rights profile to perform this procedure.

1 **Be superuser, or have the correct rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Select the zone to be modified, my-zone in this procedure.**

```
global# zonecfg -z my-zone
```

3 **Select the resource type to be changed, for example, a resource control.**

```
zonecfg:my-zone> select rctl name=zone.cpu-shares
```

4 **Remove the current value.**

```
zonecfg:my-zone:rctl> remove value (priv=privileged,limit=20,action=none)
```

5 **Add the new value.**

```
zonecfg:my-zone:rctl> add value (priv=privileged,limit=10,action=none)
```

6 **End the revised rctl specification.**

```
zonecfg:my-zone:rctl> end
```

7 **Commit the zone configuration for the zone.**

```
zonecfg:my-zone> commit
```

8 **Exit the zonecfg command.**

```
zonecfg:my-zone> exit
```

Note that even if you did not explicitly type commit at the prompt, a commit is automatically attempted when you type exit or an EOF occurs.

Committed changes made through zonecfg take effect the next time the zone is booted.

▼ How to Clear a Property Type in a Zone Configuration

Use this procedure to reset a standalone property.

1 **Be superuser, or have the correct rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Select the zone to be modified, my-zone in this procedure.**

```
global# zonecfg -z my-zone
```

3 **Clear the property to be changed, the existing pool association in this procedure.**

```
zonecfg:my-zone> clear pool
```

4 **Commit the zone configuration for the zone.**

```
zonecfg:my-zone> commit
```

5 **Exit the `zonecfg` command.**

```
zonecfg:my-zone> exit
```

Note that even if you did not explicitly type `commit` at the prompt, a `commit` is automatically attempted when you type `exit` or an EOF occurs.

Committed changes made through `zonecfg` take effect the next time the zone is booted.

▼ How to Rename a Zone

This procedure can be used to rename zones that are in either the configured state or the installed state.

You must be the global administrator in the global zone or a user with the correct rights profile to perform this procedure.

1 **Be superuser, or have the correct rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Select the zone to be renamed, `my-zone` in this procedure.**

```
global# zonecfg -z my-zone
```

3 **Change the name of the zone, for example, to `newzone`.**

```
zonecfg:my-zone> set zonename=newzone
```

4 **Commit the change.**

```
zonecfg:newzone> commit
```

5 **Exit the `zonecfg` command.**

```
zonecfg:newzone> exit
```

Committed changes made through `zonecfg` take effect the next time the zone is booted.

▼ How to Add a Dedicated Device to a Zone

The following specification places a scanning device in a non-global zone configuration.

You must be the global administrator in the global zone or a user with appropriate authorizations to perform this procedure.

1. **Become superuser, or the correct rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Add a device.**

   ```
   zonecfg:my-zone> add device
   ```

3. **Set the device match, `/dev/scsi/scanner/c3t4*` in this procedure.**

   ```
   zonecfg:my-zone:device> set match=/dev/scsi/scanner/c3t4*
   ```

4. **End the device specification.**

   ```
   zonecfg:my-zone:device> end
   ```

5. **Exit the `zonecfg` command.**

   ```
   zonecfg:my-zone> exit
   ```

▼ How to Set `zone.cpu-shares` in the Global Zone

This procedure is used to persistently set shares in the global zone.

You must be the global administrator in the global zone or a user in the global zone with the correct rights profile to perform this procedure.

1. **Be superuser, or have the correct rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Use the `zonecfg` command .**

   ```
   # zonecfg -z global
   ```

3. **Set five shares for the global zone.**

   ```
   zonecfg:global> set cpu-shares=5
   ```

4. **Exit `zonecfg`.**

   ```
   zonecfg:global> exit
   ```

Using the `zonecfg` Command to Revert or Remove a Zone Configuration

Use the `zonecfg` command described in `zonecfg`(1M) to revert a zone's configuration or to delete a zone configuration.

▼ How to Revert a Zone Configuration

While configuring a zone with the `zonecfg` utility, use the `revert` subcommand to undo a resource setting made to the zone configuration.

You must be the global administrator in the global zone or a user in the global zone with the Zone Security rights profile to perform this procedure.

1 Be superuser, or have the correct rights profile.

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 While configuring a zone called `tmp-zone`, type `info` to view your configuration:

```
zonecfg:tmp-zone> info
```

The `net` resource segment of the configuration displays as follows:

```
.
.
.
fs:
        dir: /tmp
        special: swap
        type: tmpfs
net:
        address: 192.168.0.1
        physical: eri0
device
        match: /dev/pts/*
.
.
.
```

3 Remove the net address:

```
zonecfg:tmp-zone> remove net address=192.168.0.1
```

4 Verify that the `net` entry has been removed.

```
zonecfg:tmp-zone> info

.
.
.
```

```
fs:
        dir: /tmp
        special: swap
        type: tmpfs
device
        match: /dev/pts/*
.
.
.
```

5 **Type `revert`.**

```
zonecfg:tmp-zone> revert
```

6 **Answer yes to the following question:**

```
Are you sure you want to revert (y/[n])? y
```

7 **Verify that the net address is once again present:**

```
zonecfg:tmp-zone> info

.
.
.
fs:
        dir: /tmp
        special: swap
        type: tmpfs
net:
        address: 192.168.0.1
        physical: eri0
device
        match: /dev/pts/*
.
.
.
```

▼ How to Delete a Zone Configuration

Use `zonecfg` with the `delete` subcommand to delete a zone configuration from the system.

You must be the global administrator or a user in the global zone with the Zone Security rights profile to perform this procedure.

1 **Be superuser, or have the correct rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Delete the zone configuration for the zone `a-zone` by using one of the following two methods:**

- Use the `-F` option to force the action:

```
global# zonecfg -z a-zone delete -F
```

- Delete the zone interactively by answering yes to the system prompt:

```
global# zonecfg -z a-zone delete
Are you sure you want to delete zone a-zone (y/[n])? y
```

CHAPTER 18

About Installing, Halting, Uninstalling, and Cloning Non-Global Zones (Overview)

This chapter discusses zone installation on your Oracle Solaris system. It also describes the two processes that manage the virtual platform and the application environment, `zoneadmd` and `zsched`. Information about halting, rebooting, cloning, and uninstalling zones is also provided.

The following topics are addressed in this chapter:

To clone a non-global zone, install and boot a non-global zone, or to halt or uninstall a non-global zone, see Chapter 19, "Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks)."

For information about `solaris10` branded zone installation, see Chapter 31, "Installing the `solaris10` Branded Zone."

Zone Installation and Administration Concepts

The `zoneadm` command described in the `zoneadm`(1M) man page is the primary tool used to install and administer non-global zones. Operations using the `zoneadm` command must be run from the global zone. If RBAC is in use, subcommands that make a copy of another zone require the authorization `solaris.zone.clonefrom/`*source_zone*.

The following tasks can be performed using the `zoneadm` command:

- Verify a zone
- Install a zone

- Change the state of an installed zone to incomplete
- Boot a zone, which is similar to booting a regular Oracle Solaris system
- Display information about a running zone
- Halt a zone
- Reboot a zone
- Uninstall a zone
- Relocate a zone from one point on a system to another point on the same system
- Provision a new zone based on the configuration of an existing zone on the same system
- Migrate a zone, used with the `zonecfg` command

For zone installation and verification procedures, see Chapter 19, "Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks)," and the `zoneadm`(1M) man page. Also refer to the `zoneadm`(1M) man page for supported options to the `zoneadm list` command. For zone configuration procedures, see Chapter 17, "Planning and Configuring Non-Global Zones (Tasks)," and the `zonecfg`(1M) man page. Zone states are described in "Non-Global Zone State Model" on page 198.

If you plan to produce Oracle Solaris Auditing records for zones, read "Using Oracle Solaris Auditing in Zones" on page 328 before you install non-global zones.

Zone Construction

This section applies to initial zone construction, and not to the cloning of existing zones.

After you have configured a non-global zone, you should verify that the zone can be installed safely on your system's configuration. You can then install the zone. The files needed for the zone's root file system are installed by the system under the zone's root path.

A non-global zone is installed with the limited networking configuration (`generic_limited_net.xml`). Network configuration types are described in Chapter 12, "Managing Services (Tasks)," in *System Administration Guide: Basic Administration*. The zone administrator can switch the zone to the open, traditional networking configuration (`generic_open.xml`) by using the `netservices` command. Specific services can be enabled or disabled by using SMF commands. For more information, see "Switching the Zone to a Different Networking Service Configuration or Enabling a Service" on page 291.

A successfully installed zone is ready for booting and initial login.

Data from the following are not referenced or copied when a zone is installed:

- Non-installed packages
- Data on CDs and DVDs

- Network installation images

In addition, the following types of information, if present in the global zone, are not copied into a zone that is being installed:

- New or changed users in the `/etc/passwd` file
- New or changed groups in the `/etc/group` file
- Configurations for networking services such as DHCP address assignment
- Customizations for networking services such as UUCP or sendmail
- Configurations for network services such as naming services
- New or changed `crontab`, printer, and mail files
- System log, message, and accounting files

If Oracle Solaris Auditing is used, modifications to files might be required. For more information, see "Using Oracle Solaris Auditing in Zones" on page 328.

The following features cannot be configured in a non-global zone:

- DHCP address assignment in a shared-IP zone
- SSL proxy server

The resources specified in the configuration file are added when the zone transitions from installed to ready. A unique zone ID is assigned by the system. File systems are mounted, network interfaces are set up, and devices are configured. Transitioning into the ready state prepares the virtual platform to begin running user processes. In the ready state, the `zsched` and `zoneadmd` processes are started to manage the virtual platform.

- `zsched`, a system scheduling process similar to `sched`, is used to track kernel resources associated with the zone.
- `zoneadmd` is the zones administration daemon.

A zone in the ready state does not have any user processes executing in it. The primary difference between a ready zone and a running zone is that at least one process is executing in a running zone. See the `init(1M)` man page for more information.

How Zones Are Installed

The `ipkg` brand installer supports installing the zone by using either of the following methods:

- The default repository, the solaris publisher (`http://pkg.oracle.com/solaris/release/`).

  Zones software is minimized to start. However, a full rather than minimized zone can be produced by installing an incorporation. Although not exactly the same as the SVR4 package metacluster idea of installing "everything," because IPS has different concepts, this method provides a way to install more packages during initial zone configuration. For example, the redistributable incorporation includes most of the software available in the repository and consumes approximately 9.5 gigabytes of disk space:

```
global# zoneadm -z my-zone install -e redistributable
```

To install the same set of software that is installed on a standard system, use the approximately 3–gigabyte slim_install incorporation with the -e option.

See "How to Install a Configured Zone" on page 265 for instructions.

- An image of an installed system running the Oracle Solaris release.

  The system image can be a cpio(1) archive or a pax(1) xustar archive. The cpio archive can be compressed with the gzip or bzip2 utilities. The image can also be a path to the top level of a system's root tree, or a pre-existing zone path.

  To install the zone from a system image, either the -a or -d option is required. If neither the -a or -d option is used, the zone is installed from the software repository.

The installer options are shown in the following table. See "How to Install a Configured Zone" on page 265 for example command lines.

Option	Description
-a *archive*	The path to a cpio or pax "xustar" archive of an installed system. cpio archives can be compressed using gzip or bzip. The -d and the -a options are incompatible.
-c *certificate*	The path to the certificate file needed for accessing the repository.
-d *path*	The path to the root directory of an installed system. If *path* is a hyphen (-), the zonepath is assumed to be already be populated with the system image. The -d and the -a options are incompatible.
-e *package_name*	The name of a package to install into the zone. The -e option is used to specify an additional package to install, in addition to the base set of software which is always installed into the zone. Multiple -e options can be used.
-k *key_file*	The path to the key file needed for accessing the repository.
-P *publisher=uri*	The name of a software repository publisher and its associated URI to use to install the zone instead of the default global zone's repository.
-p	Preserve system identity after installing the zone. The -p and the -u options are incompatible.
-s	Install silently. The -s and the -v options are incompatible.

Option	Description
-u	sys-unconfig the zone after installing it. The -p and the -u options are incompatible.
-v	Verbose output from the install process. The -s and the -v options are incompatible.

The zoneadmd Daemon

The zones administration daemon, zoneadmd, is the primary process for managing the zone's virtual platform. The daemon is also responsible for managing zone booting and shutting down. There is one zoneadmd process running for each active (ready, running, or shutting down) zone on the system.

The zoneadmd daemon sets up the zone as specified in the zone configuration. This process includes the following actions:

- Allocating the zone ID and starting the zsched system process
- Setting zone-wide resource controls
- Preparing the zone's devices as specified in the zone configuration
- Setting up network interfaces
- Mounting loopback and conventional file systems
- Instantiating and initializing the zone console device

Unless the zoneadmd daemon is already running, it is automatically started by zoneadm. Thus, if the daemon is not running for any reason, any invocation of zoneadm to administer the zone will restart zoneadmd.

The man page for the zoneadmd daemon is zoneadmd(1M).

The zsched Zone Scheduler

An active zone is a zone that is in the ready state, the running state, or the shutting down state. Every active zone has an associated kernel process, zsched. Kernel threads doing work on behalf of the zone are owned by zsched. The zsched process enables the zones subsystem to keep track of per-zone kernel threads.

Zone Application Environment

The zoneadm command is used to create the zone application environment.

After a non-global zone is booted for the first time, the internal configuration of the zone must be created. The internal configuration specifies a naming service to use, the default locale and time zone, the zone's root password, and other aspects of the application environment. For more information, see "Internal Zone Configuration" on page 280 and "Performing the Initial Internal Zone Configuration" on page 284. Note that the default locale and time zone for a zone can be configured independently of the global settings.

About Halting, Rebooting, and Uninstalling Zones

This section provides an overview of the procedures for halting, rebooting, uninstalling, and cloning zones.

Halting a Zone

The zoneadm halt command is used to remove both the application environment and the virtual platform for a zone. The zone is then brought back to the installed state. All processes are killed, devices are unconfigured, network interfaces are destroyed, file systems are unmounted, and the kernel data structures are destroyed.

The halt command does *not* run any shutdown scripts within the zone. To shut down a zone, see "How to Use zlogin to Shut Down a Zone" on page 290.

If the halt operation fails, see "Zone Does Not Halt" on page 359.

Rebooting a Zone

The zoneadm reboot command is used to reboot a zone. The zone is halted and then booted again. The zone ID will change when the zone is rebooted.

Zone Boot Arguments

Zones support the following boot arguments used with the zoneadm boot and reboot commands:

- -i *altinit*
- -m *smf_options*
- -s

The following definitions apply:

`-i` *altinit*
: Selects an alternative executable to be the first process. *altinit* must be a valid path to an executable. The default first process is described in `init`(1M).

`-m` *smf_options*
: Controls the boot behavior of SMF. There are two categories of options, recovery options and messages options. Message options determine the type and number of messages that displays during boot. Service options determine the services that are used to boot the system.

  Recovery options include the following:

  debug
  : Prints standard per-service output and all `svc.startd` messages to log.

  `milestone=`*milestone*
  : Boot to the subgraph defined by the given milestone. Legitimate milestones are `none`, `single-user`, `multi-user`, `multi-user-server`, and `all`.

  Message options include the following:

  `quiet`
  : Prints standard per-service output and error messages requiring administrative intervention

  `verbose`
  : Prints standard per-service output and messages providing more information.

`-s`
: Boots only to milestone `svc:/milestone/single-user:default`. This milestone is equivalent to `init` level `s`.

For usage examples, see "How to Boot a Zone" on page 269 and "How to Boot a Zone in Single-User Mode" on page 270.

For information on the Oracle Solaris service management facility (SMF) and `init` , see Chapter 11, "Managing Services (Overview)," in *System Administration Guide: Basic Administration*, `svc.startd`(1M) and `init`(1M).

Zone `autoboot` Setting

If you set the `autoboot` resource property in a zone's configuration to `true`, that zone is automatically booted when the global zone is booted. The default setting is `false`.

Note that for zones to automatically boot, the zones service `svc:/system/zones:default` must also be enabled.

See "Zones Packaging Overview" on page 301 for information on the `autoboot` setting during `pkg image-update`.

Uninstalling a Zone

The `zoneadm uninstall` command is used to uninstall all of the files under the zone's root file system. Before proceeding, the command prompts you to confirm the action, unless the `-F` (force) option is also used. Use the `uninstall` command with caution, because the action is irreversible.

About Cloning Non-Global Zones

Cloning allows you to copy an existing configured and installed zone on your system to rapidly provision a new zone on the same system. Note that at a minimum, you must reset properties and resources for the components that cannot be identical for different zones. Thus, the `zonepath` must always be changed. In addition, for a shared-IP zone, the IP addresses in any net resources must be different. For an exclusive-IP zone, the physical property of any net resources must be different.

- Cloning a zone is a faster way to install a zone.
- The new zone will include any changes that have been made to customize the source zone, such as added packages or file modifications.

When the source `zonepath` and the target `zonepath` both reside on ZFS and are in the same pool, the `zoneadm clone` command automatically uses ZFS to clone the zone. When using ZFS clone, the data is not actually copied until it is modified. Thus, the initial clone takes very little time. The `zoneadm` command takes a ZFS snapshot of the source `zonepath`, and sets up the target `zonepath`. The system names the snapshot `SUNWzone`*X*, where *X* is a unique ID used to distinguish between multiple snapshots. The `zonepath` of the destination zone is used to name the ZFS clone. A software inventory is performed so that a snapshot used at a future time can be validated by the system. To clone a source zone multiple times, the `zoneadm` command allows you to specify that an existing snapshot should be used. The system validates that the existing snapshot is usable on the target.

You cannot use manual snapshots, such as the type described in "Creating and Destroying ZFS Snapshots" in *Oracle Solaris ZFS Administration Guide*. This type of snapshot lacks the data to perform a validation.

You might want to clone a source zone many times but not want to have a new snapshot for each clone. The `-s` parameter to the `clone` subcommand allows you to specify that an existing snapshot taken from a previous clone should be used. See "How to Clone a Zone from an Existing Snapshot" on page 276.

Because a snapshot's contents represent a zone from a point in the past, it is possible that the system has been updated in some way since the snapshot was taken. The fact that the zone was upgraded could render the snapshot invalid for use as a zone on the present-day system.

Note – You can specify that a ZFS `zonepath` be copied instead of ZFS cloned, even though the source could be cloned in this way.

See "Cloning a Non-Global Zone on the Same System" on page 274 for more information.

CHAPTER 19

Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks)

This chapter describes how to install and boot a non-global zone. A method for using cloning to install a zone on the same system is also provided. Other tasks associated with installation, such as halting, rebooting, and uninstalling zones, are addressed. The procedure to completely delete a zone from a system is also included.

For general information about zone installation and related operations, see Chapter 18, "About Installing, Halting, Uninstalling, and Cloning Non-Global Zones (Overview)."

For information about `solaris10` branded zone installation and cloning, see Chapter 31, "Installing the `solaris10` Branded Zone."

Zone Installation (Task Map)

Task	Description	For Instructions
(Optional) Verify a configured zone prior to installing the zone.	Ensure that a zone meets the requirements for installation. If you skip this procedure, the verification is performed automatically when you install the zone.	"(Optional) How to Verify a Configured Zone Before It Is Installed" on page 264
Install a configured zone.	Install a zone that is in the configured state.	"How to Install a Configured Zone" on page 265
Obtain the universally unique identifier (UUID) for the zone.	This separate identifier, assigned when the zone is installed, is an alternate way to identify a zone.	"How to Obtain the UUID of an Installed Non-Global Zone" on page 267
(Optional) Transition an installed zone to the ready state.	You can skip this procedure if you want to boot the zone and use it immediately.	"(Optional) How to Transition the Installed Zone to the Ready State" on page 268

Task	Description	For Instructions
Boot a zone.	Booting a zone places the zone in the running state. A zone can be booted from the ready state or from the installed state. Note that you must perform the internal zone configuration when you log in to the zone after booting it for the first time.	"How to Boot a Zone" on page 269, "Internal Zone Configuration" on page 280, "Performing the Initial Internal Zone Configuration" on page 284
Boot a zone in single-user mode.	Boots only to milestone `svc:/milestone/single-user:default`. This milestone is equivalent to `init` level `s`. See the `init`(1M) and `svc.startd`(1M) man pages.	"How to Boot a Zone in Single-User Mode" on page 270

Installing and Booting Zones

Use the `zoneadm` command described in the `zoneadm`(1M) man page to perform installation tasks for a non-global zone. You must be the global administrator or a user with appropriate authorizations to perform the zone installation. The examples in this chapter use the zone name and zone path established in "Configuring, Verifying, and Committing a Zone" on page 238.

▼ (Optional) How to Verify a Configured Zone Before It Is Installed

You can verify a zone prior to installing it. One of the checks performed is a check for sufficient disk size. If you skip this procedure, the verification is performed automatically when you install the zone.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Verify a configured zone named `my-zone` by using the `-z` option with the name of the zone and the `verify` subcommand.**

```
global# zoneadm -z my-zone verify
```

This message regarding verification of the zone path will be displayed:

```
Warning: /zones/my-zone does not exist, so it cannot be verified.
When 'zoneadm install' is run, 'install' will try to create
/zones/my-zone, and 'verify' will be tried again,
```

```
but the 'verify' may fail if:
the parent directory of /zones/my-zone is group- or other-writable
or
/zones/my-zone overlaps with any other installed zones.
```

However, if an error message is displayed and the zone fails to verify, make the corrections specified in the message and try the command again.

If no error messages are displayed, you can install the zone.

▼ How to Install a Configured Zone

This procedure is used to install a configured non-global zone. For information on installation options, see "How Zones Are Installed" on page 255.

The zone must reside on its own ZFS dataset. Only ZFS is supported. The `zoneadm install` command automatically creates a ZFS file system (dataset) for the `zonepath` when the zone is installed. If a ZFS dataset cannot be created, the zone is not installed.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Install the configured zone `my-zone` by using the `zoneadm` command with the `install` subcommand, automatically creating a ZFS dataset for the `zonepath` ZFS. Note that the parent directory of the zone path must also be a dataset, or the file system creation will fail.**

- **Install the zone from the repository, also using the `-e` option to install additional packages:**

```
global# zoneadm -z my-zone install -c certificate_file -k key_file -P publisher=uri -e pkgA -e pkgB ...
```

- **Install the zone from an image:**

```
global# zoneadm -z my-zone install -a archive -s -u
```

- **Install the zone from a directory:**

```
global# zoneadm -z my-zone install -d path -p -v
```

- **Install the zone from an incorporation, such as the approximately 3–gigabyte slim_install incorporation:**

```
global# zoneadm -z my-zone install -e  slim_install
```

Then, remove the slim_install incorporation:

```
global# pkg -R /zones/my-zone/root uninstall slim_install
```

Removing the slim_install incorporation does not remove any of the dependent software installed by this incorporation. It removes the incorporation from the zone so that the zone matches what would be seen on a physical system.

The system will display that a ZFS dataset has been created for this zone.

You will see various messages as the files and directories needed for the zone's root file system are installed under the zone's root path.

3 **(Optional) If an error message is displayed and the zone fails to install, type the following to get the zone state:**

```
global# zoneadm -z my-zone list -v
```

- If the state is listed as configured, make the corrections specified in the message and try the `zoneadm install` command again.
- If the state is listed as incomplete, first execute this command:

  ```
  global# zoneadm -z my-zone uninstall
  ```

  Then make the corrections specified in the message, and try the `zoneadm install` command again.

4 **When the installation completes, use the `list` subcommand with the `-i` and `-v` options to list the installed zones and verify the status.**

```
global# zoneadm list -iv
```

You will see a display that is similar to the following:

```
ID  NAME     STATUS      PATH                           BRAND     IP
 0  global   running     /                              ipkg      shared
 -  my-zone  installed   /zones/my-zone                 ipkg      shared
```

Troubleshooting If a zone installation is interrupted or fails, the zone is left in the incomplete state. Use `uninstall -F` to reset the zone to the configured state.

Next Steps This zone was installed with the minimal network configuration described in Chapter 12, "Managing Services (Tasks)," in *System Administration Guide: Basic Administration* by default. You can switch to the open network configuration, or enable or disable individual services, when you log in to the zone. See "Switching the Zone to a Different Networking Service Configuration or Enabling a Service" on page 291 for details.

How to Obtain the UUID of an Installed Non-Global Zone

A universally unique identifier (UUID) is assigned to a zone when it is installed. The UUID can be obtained by using `zoneadm` with the `list` subcommand and the `-p` option. The UUID is the fifth field of the display.

- **View the UUIDs for zones that have been installed.**

  `global#` **`zoneadm list -p`**

  You will see a display similar to the following:

  ```
  0:global:running:/::ipkg:shared
  6:my-zone:running:/zones/my-zone:61901255-35cf-40d6-d501-f37dc84eb504:ipkg:shared
  ```

Example 19–1 How to Use the Zone UUID in a Command

```
global# zoneadm -z my-zone -u 61901255-35cf-40d6-d501-f37dc84eb504:shared list -v
```

If both `-u` *uuid-match* and `-z` *zonename* are present, the match is done based on the UUID first. If a zone with the specified UUID is found, that zone is used, and the `-z` parameter is ignored. If no zone with the specified UUID is found, then the system searches by the zone name.

More Information About the UUID

Zones can be uninstalled and reinstalled under the same name with different contents. Zones can also be renamed without the contents being changed. For these reasons, the UUID is a more reliable handle than the zone name.

See Also For more information, see `zoneadm`(1M) and `libuuid`(3LIB).

How to Mark an Installed Non-Global Zone Incomplete

If administrative changes on the system have rendered a zone unusable or inconsistent, it is possible to change the state of an installed zone to incomplete.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Mark the zone `testzone` incomplete.**

   `global#` **`zoneadm -z testzone mark incomplete`**

3 Use the `list` subcommand with the `-i` and `-v` options to verify the status.

```
global# zoneadm list -iv
```

You will see a display that is similar to the following:

```
ID  NAME     STATUS      PATH                           BRAND     IP
 0  global   running     /                              ipkg      shared
 -  my-zone  installed   /zones/my-zone                 ipkg      shared
 -  testzone incomplete  /zones/testzone                ipkg      shared
```

More Information

Marking a Zone Incomplete

The `-R` *root* option can be used with the `mark` and `list` subcommands of `zoneadm` to specify an alternate boot environment. See `zoneadm`(1M) for more information.

Note – Marking a zone incomplete is irreversible. The only action that can be taken on a zone marked incomplete is to uninstall the zone and return it to the configured state. See "How to Uninstall a Zone" on page 273.

▼ (Optional) How to Transition the Installed Zone to the Ready State

Transitioning into the ready state prepares the virtual platform to begin running user processes. Zones in the ready state do not have any user processes executing in them.

You can skip this procedure if you want to boot the zone and use it immediately. The transition through the ready state is performed automatically when you boot the zone.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 Be superuser, or have equivalent authorizations.

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Use the `zoneadm` command with the `-z` option, the name of the zone, which is `my-zone`, and the `ready` subcommand to transition the zone to the ready state.

```
global# zoneadm -z my-zone ready
```

3 At the prompt, use the `zoneadm list` command with the `-v` option to verify the status.

```
global# zoneadm list -v
```

You will see a display that is similar to the following:

```
ID  NAME     STATUS      PATH                          BRAND     IP
 0  global   running     /                             ipkg      shared
 1  my-zone  ready       /zones/my-zone                ipkg      shared
```

Note that the unique zone ID `1` has been assigned by the system.

▼ How to Boot a Zone

Booting a zone places the zone in the running state. A zone can be booted from the ready state or from the installed state. A zone in the installed state that is booted transparently transitions through the ready state to the running state. Zone login is allowed for zones in the running state.

Tip – Note that you perform the internal zone configuration when you initially log in to the zone. This is described in "Performing the Initial Internal Zone Configuration" on page 284.

If you plan to use an `/etc/sysidcfg` file to perform initial zone configuration, as described in "How to Use an `/etc/sysidcfg` File to Perform the Initial Zone Configuration" on page 286, create the `sysidcfg` file and place it the zone's `/etc` directory before you boot the zone.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Use the `zoneadm` command with the `-z` option, the name of the zone, which is `my-zone`, and the `boot` subcommand to boot the zone.**

   ```
   global# zoneadm -z my-zone boot
   ```

3. **When the boot completes, use the `list` subcommand with the `-v` option to verify the status.**

   ```
   global# zoneadm list -v
   ```

   You will see a display that is similar to the following:

   ```
   ID  NAME     STATUS      PATH                          BRAND     IP
    0  global   running     /                             ipkg      shared
    1  my-zone  running     /zones/my-zone                ipkg      shared
   ```

Example 19–2 Specifying Boot Arguments for Zones

Boot a zone using the `-m verbose` option:

```
global# zoneadm -z my-zone boot -- -m verbose
```

Reboot a zone using the `-m verbose` boot option:

```
global# zoneadm -z my-zone reboot -- -m verbose
```

Zone administrator reboot of the zone *my-zone*, using the `-m verbose` option:

```
my-zone# reboot -- -m verbose
```

Troubleshooting If a message indicating that the system was unable to find the netmask to be used for the IP address specified in the zone's configuration displays, see "`netmasks` Warning Displayed When Booting Zone" on page 359. Note that the message is only a warning and the command has succeeded.

▼ How to Boot a Zone in Single-User Mode

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Boot the zone in single-user mode.**

   ```
   global# zoneadm -z my-zone boot -- -s
   ```

Where to Go From Here

To log in to the zone and perform the initial internal configuration, see Chapter 20, "Non-Global Zone Login (Overview)," and Chapter 21, "Logging In to Non-Global Zones (Tasks)."

Halting, Rebooting, Uninstalling, Cloning, and Deleting Non-Global Zones (Task Map)

Task	Description	For Instructions
Halt a zone.	The halt procedure is used to remove both the application environment and the virtual platform for a zone. The procedure returns a zone in the ready state to the installed state. To cleanly shut down a zone, see "How to Use `zlogin` to Shut Down a Zone" on page 290.	"How to Halt a Zone" on page 271
Reboot a zone.	The reboot procedure halts the zone and then boots it again.	"How to Reboot a Zone" on page 272
Uninstall a zone.	This procedure removes all of the files in the zone's root file system. *Use this procedure with caution.* The action is irreversible.	"How to Uninstall a Zone" on page 273
Provision a new non-global zone based on the configuration of an existing zone on the same system.	Cloning a zone is an alternate, faster method of installing a zone. You must still configure the new zone before you can install it.	"Cloning a Non-Global Zone on the Same System" on page 274
Delete a non-global zone from the system.	This procedure completely removes a zone from a system.	"Deleting a Non-Global Zone From the System" on page 276

Halting, Rebooting, and Uninstalling Zones

▼ How to Halt a Zone

The halt procedure is used to remove both the application environment and the virtual platform for a zone. To cleanly shut down a zone, see "How to Use `zlogin` to Shut Down a Zone" on page 290.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **List the zones running on the system.**

```
global# zoneadm list -v
```

You will see a display that is similar to the following:

```
ID  NAME     STATUS      PATH                          BRAND     IP
 0  global   running     /                             ipkg      shared
 1  my-zone  running     /zones/my-zone                ipkg      shared
```

3 **Use the `zoneadm` command with the `-z` option, the name of the zone, for example, `my-zone`, and the `halt` subcommand to halt the given zone.**

```
global# zoneadm -z my-zone halt
```

4 **List the zones on the system again, to verify that `my-zone` has been halted.**

```
global# zoneadm list -iv
```

You will see a display that is similar to the following:

```
ID  NAME     STATUS      PATH                          BRAND     IP
 0  global   running     /                             ipkg      shared
 -  my-zone  installed   /zones/my-zone                ipkg      shared
```

5 **Boot the zone if you want to restart it.**

```
global# zoneadm -z my-zone boot
```

Troubleshooting If the zone does not halt properly, see "Zone Does Not Halt" on page 359 for troubleshooting tips.

▼ How to Reboot a Zone

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **List the zones running on the system.**

```
global# zoneadm list -v
```

You will see a display that is similar to the following:

```
ID  NAME     STATUS       PATH                          BRAND      IP
 0  global   running      /                             ipkg       shared
 1  my-zone  running      /zones/my-zone                ipkg       shared
```

3 **Use the `zoneadm` command with the `-z reboot` option to reboot the zone `my-zone`.**

```
global# zoneadm -z my-zone reboot
```

4 **List the zones on the system again to verify that `my-zone` has been rebooted.**

```
global# zoneadm list -v
```

You will see a display that is similar to the following:

```
ID  NAME     STATUS       PATH                          BRAND      IP
 0  global   running      /                             ipkg       shared
 2  my-zone  running      /zones/my-zone                ipkg       shared
```

Tip – Note that the zone ID for `my-zone` has changed. The zone ID generally changes after a reboot.

▼ How to Uninstall a Zone

Caution – Use this procedure with caution. The action of removing all of the files in the zone's root file system is irreversible.

The zone cannot be in the running state. The `uninstall` operation is invalid for running zones.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **List the zones on the system.**

```
global# zoneadm list -v
```

You will see a display that is similar to the following:

```
ID  NAME     STATUS       PATH                          BRAND      IP
 0  global   running      /                             ipkg       shared
 -  my-zone  installed    /zones/my-zone                ipkg       shared
```

3 Use the `zoneadm` command with the `-z uninstall` option to remove the zone `my-zone`.

You can also use the `-F` option to force the action. If this option is not specified, the system will prompt for confirmation.

```
global# zoneadm -z my-zone uninstall -F
```

Note that when you uninstall a zone that has its own ZFS file system for the `zonepath`, the ZFS file system is destroyed.

4 List the zones on the system again, to verify that `my-zone` is no longer listed.

```
global# zoneadm list -iv
```

You will see a display that is similar to the following:

```
ID  NAME     STATUS       PATH                           BRAND      IP
 0  global   running      /                              ipkg       shared
```

Troubleshooting If a zone uninstall is interrupted, the zone is left in the incomplete state. Use the `zoneadm uninstall` command to reset the zone to the configured state.

Use the `uninstall` command with caution because the action is irreversible.

Cloning a Non-Global Zone on the Same System

Cloning is used to provision a new zone on a system by copying the data from a source `zonepath` to a target `zonepath`.

When the source `zonepath` and the target `zonepath` both reside on ZFS and are in the same pool, the `zoneadm clone` command automatically uses ZFS to clone the zone. However, you can specify that the ZFS `zonepath` be copied and not ZFS cloned.

▼ How to Clone a Zone

You must configure the new zone before you can install it. The parameter passed to the `zoneadm create` subcommand is the name of the zone to clone. This source zone must be halted.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 Be superuser, or have equivalent authorizations.

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Halt the source zone to be cloned, which is `my-zone` in this procedure.

```
global# zoneadm -z my-zone halt
```

3 **Start configuring the new zone by exporting the configuration of the source zone `my-zone` to a file, for example, `master`.**

```
global# zonecfg -z my-zone export -f /zones/master
```

Note – You can also create the new zone configuration using the procedure "How to Configure the Zone" on page 238 instead of modifying an existing configuration. If you use this method, skip ahead to Step 6 after you create the zone.

---

4 **Edit the file `master`. Set different properties and resources for the components that cannot be identical for different zones. For example, you must set a new `zonepath`. For a shared-IP zone, the IP addresses in any net resources must be changed. For an exclusive-IP zone, the physical property of any net resources must be changed.**

5 **Create the new zone, `zone1`, by using the commands in the file *master*.**

```
global# zonecfg -z zone1 -f /zones/master
```

6 **Install the new zone, `zone1`, by cloning `my-zone`.**

```
global# zoneadm -z zone1 clone my-zone
```

The system displays:

```
Cloning zonepath /zones/my-zone...
```

If the source `zonepath` is on a ZFS pool, for example, `zeepool`, the system displays:

```
Cloning snapshot zeepool/zones/my-zone@SUNWzone1
Instead of copying, a ZFS clone has been created for this zone.
```

7 **List the zones on the system.**

```
ID  NAME     STATUS       PATH                           BRAND      IP
 0  global   running      /                              ipkg       shared
 -  my-zone  installed    /zones/my-zone                 ipkg       shared
 -  zone1    installed    /zones/zone1                   ipkg       shared
```

More Information

When a Source `zonepath` on a ZFS File System Is Cloned

When the `zoneadm` command clones a source `zonepath` that is on its own ZFS file system, the following actions are performed:

- The `zoneadm` command takes a software inventory.
- The `zoneadm` command takes a ZFS snapshot and names it `SUNWzone`*X*, for example, `SUNWzone`*1*.
- The `zoneadm` command uses ZFS clone to clone the snapshot.

How to Clone a Zone from an Existing Snapshot

You can clone a source zone multiple times from an existing snapshot that was originally taken when you cloned a zone.

You must be the global administratoror a user with appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Configure the zone `zone2`.**

3. **Specify that an existing snapshot be used to create `new-zone2`.**

   ```
   global# zoneadm -z zone2 clone -s zeepool/zones/my-zone@SUNWzone1 my-zone
   ```

   The system displays:

   ```
   Cloning snapshot zeepool/zones/my-zone@SUNWzone1
   ```

   The `zoneadm` command validates the software from the snapshot `SUNWzone`*1*, and clones the snapshot.

4. **List the zones on the system.**

   ```
   ID  NAME     STATUS      PATH                           BRAND    IP
    0  global   running     /                              ipkg     shared
    -  my-zone  installed   /zeepool/zones/my-zone         ipkg     shared
    -  zone1    installed   /zeepool/zones/zone1           ipkg     shared
    -  zone2    installed   /zeepool/zones/zone2           ipkg     shared
   ```

Deleting a Non-Global Zone From the System

The procedure described in this section completely deletes a zone from a system.

How to Remove a Non-Global Zone

1. **Shut down the zone `my-zone`.**

   ```
   global# zlogin my-zone shutdown
   my-zone
   ```

2. **Remove the root file system for `my-zone`.**

   ```
   global# zoneadm -z my-zone uninstall -F
   ```

3 **Delete the configuration for `my-zone`.**

```
global# zonecfg -z my-zone delete -F
```

4 **List the zones on the system, to verify that `my-zone` is no longer listed.**

```
global# zoneadm list -iv
```

You will see a display that is similar to the following:

```
ID  NAME     STATUS       PATH                           BRAND      IP
 0  global   running      /                              ipkg       shared
```

CHAPTER 20

Non-Global Zone Login (Overview)

This chapter discusses logging in to zones from the global zone.

The following topics are covered in this chapter:

For procedures and usage information, see Chapter 21, "Logging In to Non-Global Zones (Tasks)."

zlogin Command

If RBAC is in use, access to the zone console requires the authorization `solaris.zone.manage/`*zonename*. A specific *zonename* suffix, preceded by the slash character (`/`), is optional. When omitted, the authorization matches any zone.

After you install a zone, you must log in to the zone to complete its application environment. You might log in to the zone to perform administrative tasks as well. Unless the `-C` option is used to connect to the zone console, logging in to a zone using `zlogin` starts a new task. A task cannot span two zones.

The `zlogin` command is used to log in from the global zone to any zone that is in the running state or the ready state.

Note – Only the zlogin command with the -C option can be used to log in to a zone that is not in the running state.

As described in "How to Use Non-Interactive Mode to Access a Zone" on page 288, you can use the zlogin command in non-interactive mode by supplying a command to run inside a zone. However, the command or any files the command acts upon cannot reside on NFS. The command will fail if any of its open files or any portion of its address space resides on NFS. The address space includes the command executable itself and the command's linked libraries.

The zlogin command can only be used by the global administrator or a user with appropriate authorizations, operating in the global zone. See the zlogin(1) man page for more information.

Internal Zone Configuration

After installation, the zone is in an unconfigured state. The zone does not have an internal configuration for naming services, its locale and time zone have not been set, and various other configuration tasks have not been performed. Therefore, the sysidtool programs are run the first time a zone is booted. For more information, see the sysidtool(1M) man page.

Two methods are available for performing the required configuration:

- Zone console login, which initiates a series of questions from the system. Be prepared to respond to the following:
  - Language
  - Type of terminal being used
  - Host name
  - Security policy (Kerberos or standard UNIX)
  - Naming service type (None is a valid response)
  - Naming service domain
  - Name server
  - Default time zone
  - Root password

  The procedure is described in "Performing the Initial Internal Zone Configuration" on page 284.
- An /etc/sysidcfg file, which you can create and place inside the zone before you boot the zone for the first time. See the sysidcfg(4) man page for more information.

Non-Global Zone Login Methods

This section describes the methods you can use to log in to a zone.

Zone Console Login

Each zone maintains a virtual console, `/dev/console`. Performing actions on the console is referred to as console mode. Console login to a zone is available when the zone is in the installed state. The zone console is closely analogous to a serial console on a system. Connections to the console persist across zone reboots. To understand how console mode differs from a login session such as `telnet`, see "Remote Login" on page 282.

The zone console is accessed by using the `zlogin` command with the `-C` option and the *zonename*. The zone does not have to be in the running state.

Processes inside the zone can open and write messages to the console. If the `zlogin -C` process exits, another process can then access the console.

If role-based access control (RBAC) is in use, access to the zone console requires the authorization `solaris.zone.manage/`*zonename*. A specific *zonename* suffix, preceded by the slash character (`/`), is optional. When omitted, the authorization matches any zone.

User Login Methods

To log in to the zone with a user name, use the `zlogin` command with the `-l` option, the user name, and the *zonename*. For example, the administrator of the global zone can log in as a normal user in the non-global zone by specifying the `-l` option to `zlogin`:

```
global# zlogin -l user zonename
```

To log in as user `root`, use the `zlogin` command without options.

Failsafe Mode

If a login problem occurs and you cannot use the `zlogin` command or the `zlogin` command with the `-C` option to access the zone, an alternative is provided. You can enter the zone by using the `zlogin` command with the `-S` (safe) option. Only use this mode to recover a damaged zone when other forms of login are not succeeding. In this minimal environment, it might be possible to diagnose why the zone login is failing.

Remote Login

The ability to remotely log in to a zone is dependent on the selection of network services that you establish. By default, a non-global zone is installed with the limited networking configuration (`/var/svc/profile/generic_limited_net.xml`), and only the `ssh` login is enabled. Logins through `rlogin` and `telnet` can be added if needed, either by using the `netservices` command to switch the zone to the open networking configuration or by enabling the services using SMF.

For more information about changing the network profile or using SMF commands to add services to zones, see "Switching the Zone to a Different Networking Service Configuration or Enabling a Service" on page 291. For more information about login commands, see `rlogin(1)`, `ssh(1)`, and `telnet(1)`

Interactive and Non-Interactive Modes

Two other methods for accessing the zone and for executing commands inside the zone are also provided by the `zlogin` command. These methods are interactive mode and non-interactive mode.

Interactive Mode

In interactive mode, a new pseudo-terminal is allocated for use inside the zone. Unlike console mode, in which exclusive access to the console device is granted, an arbitrary number of `zlogin` sessions can be open at any time in interactive mode. Interactive mode is activated when you do not include a command to be issued. Programs that require a terminal device, such as an editor, operate correctly in this mode.

If role-based access control (RBAC) is in use, for interactive logins, the authorization `solaris.zone.login/`*zonename* for the zone is required. Password authentication takes place in the zone.

Non-Interactive Mode

Non-interactive mode is used to run shell-scripts which administer the zone. Non-interactive mode does not allocate a new pseudo-terminal. Non-interactive mode is enabled when you supply a command to be run inside the zone.

For non-interactive logins, or to bypass password authentication, the authorization `solaris.zone.manage/`*zonename* is required.

CHAPTER 21

Logging In to Non-Global Zones (Tasks)

This chapter provides procedures for completing the configuration of an installed zone, logging into a zone from the global zone, and shutting down a zone. This chapter also shows how to use the `zonename` command to print the name of the current zone.

For an introduction to the zone login process, see Chapter 20, "Non-Global Zone Login (Overview)."

Initial Zone Boot and Zone Login Procedures (Task Map)

Task	Description	For Instructions
Perform the internal configuration.	Log in to the zone console or use an `/etc/sysidcfg` file to perform the initial zone configuration.	"Performing the Initial Internal Zone Configuration" on page 284
Log in to the zone.	You can log into a zone through the console, by using interactive mode to allocate a pseudo-terminal, or by supplying a command to be run in the zone. Supplying a command to be run does not allocate a pseudo-terminal. You can also log in by using failsafe mode when a connection to the zone is denied.	"Logging In to a Zone" on page 287
Exit a non-global zone.	Disconnect from a non-global zone.	"How to Exit a Non-Global Zone" on page 289
Shut down a zone.	Shut down a zone by using the `shutdown` utility or a script.	"How to Use `zlogin` to Shut Down a Zone" on page 290

Task	Description	For Instructions
Print the zone name.	Print the zone name of the current zone.	"Printing the Name of the Current Zone" on page 292

Performing the Initial Internal Zone Configuration

You must configure the zone using one of the following methods:

- Log into the zone and configure it as described in "Internal Zone Configuration" on page 280.
- Configure the zone using an `/etc/sysidcfg` file as described in "How to Use an `/etc/sysidcfg` File to Perform the Initial Zone Configuration" on page 286.

Tip – After you have performed the internal configuration, it is a good idea to make a copy of the non-global zone's configuration. You can use this backup to restore the zone in the future. As superuser or using an equivalent role, print the configuration for the zone `my-zone` to a file. This example uses a file named `my-zone.config`.

```
global# zonecfg -z my-zone export > my-zone.config
```

See "How to Restore an Individual Non-Global Zone" on page 355 for more information.

▼ How to Log In to the Zone Console to Perform the Internal Zone Configuration

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Use the `zlogin` command with the `-C` option and the name of the zone, `my-zone` in this procedure.**

   ```
   global# zlogin -C my-zone
   ```

3. **From another terminal window, boot the zone.**

   ```
   global# zoneadm -z my-zone boot
   ```

   You will see a display similar to the following in the `zlogin` window:

   ```
   [NOTICE: Zone booting up]
   ```

4 The first time you log in to the console, you are prompted to answer a series of questions. Your screen will look similar to this:

```
SunOS Release 5.11 Version Generic 64-bit
Copyright 1983-2007 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Hostname: my-zone
Loading smf(5) service descriptions: 114/114
Select a Language

     1. English
     2. es
     2. fr

Please make a choice (1 - 3), or press h or ? for help:

Select a Locale

      1. English (C - 7-bit ASCII)
      2. Canada (English) (UTF-8)
      4. U.S.A. (UTF-8)
      5. U.S.A. (en_US.ISO8859-1)
      6. U.S.A. (en_US.ISO8859-15)
      7. Go Back to Previous Screen

Please make a choice (1 - 7), or press h or ? for help:

What type of terminal are you using?
      1) ANSI Standard CRT
      2) DEC VT52
      3) DEC VT100
      4) Heathkit 19
      5) Lear Siegler ADM31
      6) PC Console
      7) Sun Command Tool
      8) Sun Workstation
      9) Televideo 910
      10) Televideo 925
      11) Wyse Model 50
      12) X Terminal Emulator (xterms)
Type the number of your choice and press Return:
12
.
.
.
```

For the complete list of questions you must answer, see "Internal Zone Configuration" on page 280.

5 (Optional) If you are not using two windows as described in step 3, you might have missed the initial prompt for configuration information. If you see the following system message at zone login instead of a prompt:

[connected to zone *zonename* console]

Press Return to display the prompt again.

If you enter an incorrect response and try to restart the configuration, you might experience difficulty when you attempt the process again. This occurs because the `sysidtools` can store your previous responses.

If this happens, use the following workaround from the global zone to restart the configuration process.

```
global# zlogin -S zonename /usr/sbin/sys-unconfig
```

For more information on the `sys-unconfig` command, see the `sys-unconfig`(1M) man page.

How to Use an `/etc/sysidcfg` File to Perform the Initial Zone Configuration

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **From the global zone, change directories to the non-global zone's `/etc` directory:**

   ```
   global# cd /zones/my-zone/root/etc
   ```

3. **Create the `sysidcfg` file and place it in this directory.**

   The file will look similar to the following:

   - **For a shared-IP zone:**

     ```
     system_locale=C
     terminal=xterms
     network_interface=primary {
                 hostname=my-zone
     }
     security_policy=NONE
     name_service=NIS {
                 domain_name=special.example.com
                 name_server=bird(192.168.112.3)
     }
     nfs4_domain=domain
     timezone=US/Central
     root_password=m4qtoWN
     ```

   - **For an exclusive-IP zone with a static IP configuration:**

     ```
     system_locale=C
     terminal=xterms
     network_interface=primary {
              hostname=my-zone
     ```

```
                 default_route=10.10.10.1
                 ip_address=10.10.10.13
                 netmask=255.255.255.0
}
nfs4_domain=domain
timezone=US/Central
root_password=m4qtoWN
```

- **For an exclusive-IP zone with DHCP and IPv6 option:**

```
system_locale=C
terminal=dtterm
network_interface=primary {
                dhcp protocol_ipv6=yes
}
security_policy=NONE
name_service=DNS {
        domain_name=example.net
        name_server=192.168.224.11,192.168.224.33
}
nfs4_domain=domain
timezone=US/Central
root_password=m4qtoWN
```

4 **Boot the zone.**

See Also See the sysidcfg(4) man page for more information.

Logging In to a Zone

Use the `zlogin` command to log in from the global zone to any zone that is running or in the ready state. See the `zlogin(1)` man page for more information.

You can log in to a zone in various ways, as described in the following procedures. You can also log in remotely, as described in "Remote Login" on page 282.

▼ How to Log In to the Zone Console

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Use the `zlogin` command with the `-C` option and the name of the zone, for example, `my-zone`.**

```
global# zlogin -C my-zone
```

Note – If you start the `zlogin` session immediately after issuing the `zoneadm boot` command, boot messages from the zone will display:

```
[NOTICE: Zone booting up]
SunOS Release 5.11 Version snv_145 64-bit
Copyright (c) 1983, 2010, Oracle and/or its affiliates. All rights reserved.
Hostname: my-zone
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
```

3. **When the zone console displays, log in as `root`, press Return, and type the root password when prompted.**

```
my-zone console login: root
Password:
```

▼ How to Use Interactive Mode to Access a Zone

In interactive mode, a new pseudo-terminal is allocated for use inside the zone.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **From the global zone, log in to the zone, for example, `my-zone`.**

```
global# zlogin my-zone
```

   Information similar to the following will display:

```
[Connected to zone 'my-zone' pts/2]
Last login: Wed Jul  3 16:25:00 on console
```

3. **Type `exit` to close the connection.**

   You will see a message similar to the following:

```
[Connection to zone 'my-zone' pts/2 closed]
```

▼ How to Use Non-Interactive Mode to Access a Zone

Non-interactive mode is enabled when the user supplies a command to be run inside the zone. Non-interactive mode does not allocate a new pseudo-terminal.

Note that the command or any files that the command acts upon cannot reside on NFS.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **From the global zone, log in to the `my-zone` zone and supply a command name.**

The command `zonename` is used here.

```
global# zlogin my-zone zonename
```

You will see the following output:

```
my-zone
```

How to Exit a Non-Global Zone

- **To disconnect from a non-global zone, use one of the following methods.**
  - **To exit the zone non-virtual console:**

    ```
    zonename# exit
    ```

  - **To disconnect from a zone virtual console, use the tilde (~) character and a period:**

    ```
    zonename# ~.
    ```

    Your screen will look similar to this:

    ```
    [Connection to zone 'lx-zone' pts/6 closed]
    ```

See Also For more information about `zlogin` command options, see the `zlogin(1)` man page.

How to Use Failsafe Mode to Enter a Zone

When a connection to the zone is denied, the `zlogin` command can be used with the `-S` option to enter a minimal environment in the zone.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **From the global zone, use the `zlogin` command with the `-S` option to access the zone, for example, `my-zone`.**

```
global# zlogin -S my-zone
```

How to Use `zlogin` to Shut Down a Zone

Note – Running `init 0` in the global zone to cleanly shut down a Oracle Solaris system also runs `init 0` in each of the non-global zones on the system. Note that `init 0` does not warn local and remote users to log off before the system is taken down.

Use this procedure to cleanly shut down a zone. To halt a zone without running shutdown scripts, see "How to Halt a Zone" on page 271.

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Log in to the zone to be shut down, for example, `my-zone`, and specify `shutdown` as the name of the utility and init 0 as the state.**

```
global# zlogin my-zone shutdown -i 0
```

Your site might have its own shutdown script, tailored for your specific environment.

Note – You cannot use the `shutdown` command to place the zone in single-user state at this time. See 6214427 for more information.

More Information

Using `shutdown` in Non-Interactive Mode

You cannot use the `shutdown` command in non-interactive mode to place the zone in single-user state at this time. See 6214427 for more information.

You can use an interactive login as described in "How to Use Interactive Mode to Access a Zone" on page 288.

Switching the Zone to a Different Networking Service Configuration or Enabling a Service

A zone is installed with the minimal networking service configuration described in Chapter 12, "Managing Services (Tasks)," in *System Administration Guide: Basic Administration*. You can switch the zone to the open networking service configuration. You can also enable or disable individual services in the zone.

▼ How to Switch the Zone to the Open Networking Service Configuration

1. **From the global zone, log in to the zone, for example, `my-zone`.**

   ```
   global# zlogin my-zone
   ```

2. **Run the `netservices` command to switch the zone to the traditional open networking configuration.**

   ```
   my-zone# /usr/sbin/netservices open
   ```

   You will see a display similar to the following. Respond *y* to the prompt to restart `dtlogin`.

   ```
   restarting syslogd
   restarting sendmail
   dtlogin needs to be restarted. Restart now? [Y] y
   restarting dtlogin
   ```

▼ How to Enable a Specific Service in a Zone

1. **From the global zone, log in to the zone, for example, `my-zone`.**

   ```
   global# zlogin my-zone
   ```

2. **Run the `svcadm` command to enable `rlogin`.**

   ```
   my-zone# svcadm enable svc:/network/login:rlogin
   ```

3. **List the services to verify that `rlogin` is enabled.**

   ```
   my-zone# svcs -a
   .
   .
   .
   online      14:01:08 svc:/network/login:rlogin
   .
   .
   .
   ```

Printing the Name of the Current Zone

The zonename command described in the zonename(1) man page prints the name of the current zone. The following example shows the output when zonename is used in the global zone.

```
# zonename
global
```

CHAPTER 22

Moving and Migrating Non-Global Zones (Tasks)

This chapter describes how to:

- Move an existing non-global zone to a new location on the same machine.
- Validate what will happen in a non-global zone migration before the actual migration is performed.
- Migrate an existing non-global zone to a new machine.

This information also applies to moving and migrating `solaris10` branded zones. For information on `solaris10` branded zones, see Part III, "Oracle Solaris 10 Zones."

Moving a Non-Global Zone

This procedure is used to move the zone to a new location on the same system by changing the `zonepath`. The zone must be halted. The new `zonepath` must be on a local file system. The normal `zonepath` criteria described in "Resource and Property Types" on page 221 apply.

▼ How to Move a Zone

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Halt the zone to be moved, `db-zone` in this procedure.**

```
global# zoneadm -z db-zone halt
```

3 Use the `zoneadm` command with the `move` subcommand to move the zone to a new `zonepath`, `/zones/db-zone`.

```
global# zoneadm -z db-zone move /zones/db-zone
```

4 Verify the path.

```
ID  NAME     STATUS       PATH                           BRAND    IP
 0  global   running      /                              ipkg     shared
 -  my-zone  installed    /zones/my-zone                 ipkg     shared
 -  db-zone  installed    /zones/db-zone                 ipkg     shared
```

Migrating a Non-Global Zone to a Different Machine

Note that you can do a trial run of a zone migration before you actually move the zone to a different machine. For more information, see "About Validating a Zone Migration Before the Migration Is Performed" on page 298.

About Migrating a Zone

The `zonecfg` and `zoneadm` commands can be used to migrate an existing non-global zone from one system to another. The zone is halted and detached from its current host. The `zonepath` is moved to the target host, where it is attached.

The following requirements apply to zone migration:

- The global zone on the target system must be running the same Oracle Solaris release as the original source host.
- To ensure that the zone will run properly, the target system must have the same or later versions of the required operating system packages as those installed on the original source host.

  Other packages, such as those for third-party products, can be different.
- If the new host has later versions of the zone-dependent packages, using `zoneadm attach` with the `-u` option updates those packages within the zone to match the new host. The update on attach software looks at the zone that is being migrated and determines which packages must be updated to match the new host. Only those packages are updated. The rest of the packages, and their associated patches, can vary from zone to zone.
- The host and target systems must have the same machine class unless the `-u` option, which enables automatic migration between `sun4u` and `sun4v` machine classes, is used.

To verify the Oracle Solaris release and the machine architecture, type:

```
#uname -m
```

The `zoneadm detach` process creates the information necessary to attach the zone on a different system. The `zoneadm attach` process verifies that the target machine has the correct configuration to host the zone.

Because there are several ways to make the `zonepath` available on the new host, the actual movement of the `zonepath` from one system to another is a manual process that is performed by the global administrator.

When attached to the new system, the zone is in the installed state.

▼ How to Migrate A Non-Global Zone

You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Halt the zone to be migrated, `my-zone` in this procedure.**

   ```
   host1# zoneadm -z my-zone halt
   ```

3. **Detach the zone.**

   ```
   host1# zoneadm -z my-zone detach
   ```

   The detached zone is now in the configured state.

4. **Move the `zonepath` for `my-zone` to the new host.**

   See "How to Move the `zonepath` to a new Host" on page 297 for more information.

5. **On the new host, configure the zone.**

   ```
   host2# zonecfg -z my-zone
   ```

   You will see the following system message:

   ```
   my-zone: No such zone configured
   Use 'create' to begin configuring a new zone.
   ```

6. **To create the zone `my-zone` on the new host, use the `zonecfg` command with the `-a` option and the `zonepath` on the new host.**

   ```
   zonecfg:my-zone> create -a /zones/my-zone
   ```

7. **(Optional) View the configuration.**

   ```
   zonecfg:my-zone> info
   zonename: my-zone
   zonepath: /zones/my-zone
   ```

```
autoboot: false
pool:
net:
         address: 192.168.0.90
         physical: bge0
```

8 Make any required adjustments to the configuration.

For example, the network physical device is different on the new host, or devices that are part of the configuration might have different names on the new host.

```
zonecfg:my-zone> select net physical=bge0
zonecfg:my-zone:net> set physical=e1000g0
zonecfg:my-zone:net> end
```

9 Commit the configuration and exit.

```
zonecfg:my-zone> commit
zonecfg:my-zone> exit
```

10 Attach the zone on the new host using one of the following methods.

- **Attach the zone from the archive that was created on the source system, with the archive transferred into the `/zones` directory on the destination system:**

  ```
  host2# zoneadm -z my-zone attach -a /zones/my-zone.cpio.gz -u
  ```

- **If you've simply detached the zone on this system and now want to re-attach it with a validation step:**

  ```
  host2# zoneadm -z my-zone attach
  ```

  The system administrator is notified of required actions to be taken if either or both of the following conditions are present:

  - Required packages and patches are not present on the new machine.
  - The software levels are different between machines.

- **Attach the zone with a validation check and update the zone to match a host running later versions of the dependent packages or having a different machine class upon attach.**

  ```
  host2# zoneadm -z my-zone attach -u
  ```

- **Force the attach operation without performing the validation.**

  ```
  host2# zoneadm -z my-zone attach -F
  ```

Caution – The `-F` option allows you to force the `attach` with no validation performed. This is useful in certain cases, such as in a clustered environment or for backup and restore operations, but it does require that the system be properly configured to host the zone. An incorrect configuration could result in undefined behavior later.

How to Move the `zonepath` to a new Host

There are many ways to create an archive of the `zonepath`. For example, you can use the `cpio` or `pax` commands described in the `cpio(1)`) and `pax(1)` man pages.

There are also several ways to transfer the archive to the new host. The mechanism used to transfer the `zonepath` from the source host to the destination depends on the local configuration. In some cases, such as a SAN, the `zonepath` data might not actually move. The SAN might simply be reconfigured so the `zonepath` is visible on the new host. In other cases, the `zonepath` might be written to tape, and the tape mailed to a new site.

For these reasons, this step is not automated. The system administrator must choose the most appropriate technique to move the `zonepath` to the new host.

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Move the `zonepath` to the new host. You can use the method described in this procedure, or use another method of your choice.**

Example 22–1 Archiving and Moving the `zonepath` Using the `tar` Command, and Attaching the Zone

1. Create a `tar` file of the `zonepath` on `host1` and transfer it to `host2` by using the `sftp` command.

```
host1# cd /zones
host1# tar cf my-zone.tar my-zone
host1# sftp host2
Connecting to host2...
Password:
sftp> cd /zones
sftp> put my-zone.tar
Uploading my-zone.tar to /zones/my-zone.tar
sftp> quit
```

2. On `host2`, attach the zone:

```
host2# zoneadm -z my-zone attach -a /zones/my-zone.tar -u
```

For more information, see `sftp(1)` and `tar(1)`.

Example 22–2 Archiving the `zonepath` Using the `cpio` and Compressing the Archive Using `gzip`

This is an alternative to using the `tar` command as shown in Example 22-1.

```
host1# find my-zone -print | cpio -oP@/ | gzip > my-zone.cpio.gz
```

Next Steps If you have used the `-a` option instead of reconfiguring a SAN, then the `zonepath` data will still be visible on the source host even though the zone is now in the configured state. You can either manually remove the `zonepath` from the source host after you have finished moving the data to the new host, or you can reattach the zone to the source host and use the `zoneadm uninstall` command to remove the `zonepath`.

About Validating a Zone Migration Before the Migration Is Performed

You can perform a trial run before the zone is moved to the new machine by using the "no execute" option, `-n`.

The `zoneadm detach` subcommand is used with the `-n` option to generate a manifest on a running zone without actually detaching the zone. The state of the zone on the originating system is not changed. The zone manifest is sent to `stdout`. The global administrator or a user with appropriate authorizations can direct this output to a file or pipe it to a remote command to be immediately validated on the target host. The `zoneadm attach` subcommand is used with the `-n` option to read this manifest and verify that the target machine has the correct configuration to host the zone without actually doing an attach.

The zone on the target system does *not* have to be configured on the new host before doing a trial-run attach.

▼ How to Validate a Zone Migration Before the Migration Is Performed

You must be the global administrator or a user with the Zone Security role in the global zone to perform this procedure.

1. **Be superuser, or assume an equivalent role.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Use one of the following methods.**

   - **Generate the manifest on a source host named `my-zone` and pipe the output to a remote command that will immediately validate the target host:**

     ```
     global# zoneadm -z my-zone detach -n | ssh remotehost zoneadm attach -n -
     ```

     The hyphen (–) at the end of the line specifies `stdin` for the path.

   - **Generate the manifest on a source host named `my-zone` and direct the output to a file:**

     ```
     global# zoneadm -z my-zone detach -n
     ```

Copy the manifest to the new host system as described in [“How to Move the `zonepath` to a new Host” on page 297](#)**, and perform the validation:**

```
global# zoneadm attach -n path_to_manifest
```

The path can be – to specify `stdin`.

Migrating a Zone From a Machine That Is not Usable

A machine that hosts an Oracle Solaris zone can become unusable. However, if the storage that the zone lives on, such as a SAN, is still usable, it might still be possible to migrate the zone to a new host successfully. You can move the `zonepath` for the zone to the new host. In some cases, such as a SAN, the `zonepath` data might not actually move. The SAN might simply be re-configured so the `zonepath` is visible on the new host. Since the zone was not properly detached, you will have to first create the zone on the new host using the `zonecfg` command. Once this has been done, attach the zone on the new host.

The procedure for this task is described in steps 4 through 8 of “How to Migrate A Non-Global Zone” on page 295. Also see “How to Move the `zonepath` to a new Host” on page 297.

CHAPTER 23

About Packages on an Oracle Solaris 11 Express System With Zones Installed

Image Packaging System (IPS) packages are supported for the Oracle Solaris 11 Express release. This chapter discusses maintaining the operating system by using IPS packaging when zones are installed.

For information about SVR4 packaging and patching used in `solaris10` and `native` zones, see "Chapter 25, About Packages on an Solaris System With Zones Installed (Overview)" and "Chapter 26, Adding and Removing Packages and Patches on a Solaris System With Zones Installed (Tasks)" in *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones*. This is the Oracle Solaris 10 version of the guide.

Image Packaging System Software on Systems Running the Oracle Solaris 11 Express Release

This chapter provides information about adding packages to the installed non-global zone. Information about removing packages is also included. The material in this chapter supplements the existing Oracle Solaris installation and packaging documentation. For more information, see the following documents.

- *Oracle Solaris 11 Express Image Packaging System Guide*
- *System Administration Guide: Basic Administration*

Zones Packaging Overview

The `solaris` packaging repository is used in administering the zones environment.

Currently, non-global zones require manual syncing. The zones don't automatically update when you use `pkg image-update` to upgrade the system to a new version of Oracle Solaris. You must manually update the zones after rebooting to keep them in sync with the global zone.

Note – Until `pkg image-update` in zones changes, you can use `zoneadm detach` and `attach -u` as a workaround. Run `pkg image-update` and reboot into the new image, then detach the zone and use `zoneadm attach` with the `-u` option. See "About Migrating a Zone" on page 294 for more information on these commands.

If the zone is set `autoboot=true`, then this setting should be changed to `autoboot=false` before the `pkg image-update`. Once the new BE is booted and the zones are synced up to the global zone, then `autoboot` can be turned back on (reset to `true`).

The Image Packaging System (IPS), described in `pkg`(5), is a framework that provides for software lifecycle management such as installation, upgrade, and removal of packages. IPS can be used to create software packages, create and manage packaging repositories, and mirror existing packaging repositories.

After an initial installation of the Oracle Solaris operating system, you can install additional software applications from a packaging repository through the Image Packaging System CLI and GUI (Package Manager) clients.

After you have installed the packages on your system, the IPS clients can be used to search, upgrade, and manage them. The IPS clients can be also used to upgrade an entire system to a new release of Oracle Solaris, create and manage repositories, and mirror an existing repository.

If the system on which IPS is installed can access the Internet, then the clients can access and install software from the Oracle Solaris 11 Package Repository (default `solaris` publisher), `http://pkg.oracle.com/solaris/release/`.

The zone administrator can use the packaging tools to administer any software installed in a non-global zone, within the limits described in this document.

The following general principles apply when zones are installed:

- The global administrator or a user with appropriate authorizations can administer the software on every zone on the system.
- The root file system for a non-global zone can be administered from the global zone by using the Oracle Solaris packaging tools. The Oracle Solaris packaging tools are supported within the non-global zone for administering co-packaged (bundled), standalone (unbundled), or third-party products.
- The packaging tools work in a zones-enabled environment. The tools allow a package to also be installed in a non-global zone.

Note – While certain package operations are performed, a zone is temporarily locked to other operations of this type. The system might also confirm a requested operation with the administrator before proceeding.

About Packages and Zones

The `ipkg` zone uses the branded zones framework described in `brands(5)` to run zones installed with the same software as is installed in the global zone. The system software must always be in sync with the global zone when using an `ipkg` brand. The system software packages within the zone are managed using the image packaging system.

The various components within a package are specifically tagged to only be installed in either a global zone or a non-global zone. A given package can contain a file that is tagged so that it won't be installed into a non-global zone.

Only a subset of the Oracle Solaris packages installed in the global zone are completely replicated when a non-global zone is installed. For example, many packages that contain the Oracle Solaris kernel are not needed in a non-global zone. All non-global zones implicitly share the same kernel from the global zone.

To learn about IPS packaging on an Oracle Solaris system, see *Oracle Solaris 11 Express Image Packaging System Guide*.

How Zone State Affects Package Operations

The following table describes what will happen when packaging commands are used on a system with non-global zones in various states.

Zone State	Effect on Package Operations
Configured	Package tools can be run. No software has been installed yet.
Installed	Package tools can be run. Note that immediately after `zoneadm -z` *zonename* `install` has completed, the zone is also moved to the installed state.
Ready	Package tools can be run.
Running	Package tools can be run.

Zone State	Effect on Package Operations
Incomplete	A zone being installed or removed by `zoneadm`. Package tools cannot be used. The zone is not in an appropriate state for using the tools.

About Adding Packages in Systems With Zones Installed

On the Oracle Solaris 11 Express release, use the `pkg install` command.

```
# pkg install SUNWphp524-mysql
```

Using `pkg` in the Global Zone

The `pkg install` command is used in the global zone to add the package to the global zone only. The package is not propagated to any other zones. For more information, see the *Oracle Solaris 11 Express Image Packaging System Guide*.

Using the `pkg install` Command in a Non-Global Zone

The `pkg install` command is used by the zone administrator in the non-global zone to add the package to the non-global zone only. To add a package in a specified non-global zone, execute the `pkg install` command as the zone administrator. For more information, see the *Oracle Solaris 11 Express Image Packaging System Guide*.

Package dependencies are handled automatically in IPS.

For more information, see the *Oracle Solaris 11 Express Image Packaging System Guide*.

About Removing Packages in Zones

On the Oracle Solaris 11 Express release, use the `pkg uninstall` command to remove packages on a system with zones installed.

```
# pkg uninstall SUNWphp524-mysql
```

Package Information Query

On the Oracle Solaris 11 Express release, use the `pkg info` command to query the software package database on a system with zones installed.

The command can be used in the global zone to query the software package database in the global zone only. The command can be used in a non-global zone to query the software package database in the non-global zone only.

CHAPTER 24

Oracle Solaris Zones Administration (Overview)

This chapter covers these general zone administration topics:

For information on `solaris10` branded zones, see Part III, "Oracle Solaris 10 Zones."

Global Zone Visibility and Access

The global zone acts as both the default zone for the system and as a zone for system-wide administrative control. There are administrative issues associated with this dual role. Since applications within the zone have access to processes and other system objects in other zones, the effect of administrative actions can be wider than expected. For example, service shutdown scripts often use `pkill` to signal processes of a given name to exit. When such a script is run from the global zone, all such processes in the system will be signaled, regardless of zone.

The system-wide scope is often needed. For example, to monitor system-wide resource usage, you must view process statistics for the whole system. A view of just global zone activity would miss relevant information from other zones in the system that might be sharing some or all of the system resources. Such a view is particularly important when system resources such as CPU are not strictly partitioned using resource management facilities.

Thus, processes in the global zone can observe processes and other objects in non-global zones. This allows such processes to have system-wide observability. The ability to control or send signals to processes in other zones is restricted by the privilege `PRIV_PROC_ZONE`. The privilege is similar to `PRIV_PROC_OWNER` because the privilege allows processes to override the restrictions placed on unprivileged processes. In this case, the restriction is that unprivileged processes in the global zone cannot signal or control processes in other zones. This is true even when the user IDs of the processes match or the acting process has the `PRIV_PROC_OWNER` privilege. The `PRIV_PROC_ZONE` privilege can be removed from otherwise privileged processes to restrict actions to the global zone.

For information about matching processes by using a `zoneidlist`, see the `pgrep(1)` `pkill(1)` man pages.

Process ID Visibility in Zones

Only processes in the same zone will be visible through system call interfaces that take process IDs, such as the `kill` and `priocntl` commands. For information, see the `kill(1)` and the `priocntl(1)` man pages.

System Observability in Zones

The `ps` command has the following modifications:

- The `-o` option is used to specify output format. This option allows you to print the zone ID of a process or the name of the zone in which the process is running.
- The `-z` *zonelist* option is used to list only processes in the specified zones. Zones can be specified either by zone name or by zone ID. This option is only useful when the command is executed in the global zone.

- The -Z option is used to print the name of the zone associated with the process. The name is printed under the column heading ZONE.

For more information, see the ps(1) man page.

A -z *zonename* option has been added to the following Oracle Solaris utilities. You can use this option to filter the information to include only the zone or zones specified.

- ipcs (see the ipcs(1) man page)
- pgrep (see the pgrep(1) man page)
- ptree (see the proc(1) man page)
- prstat (see the prstat(1M) man page)

See Table 24–5 for the full list of changes made to commands.

Reporting Active Zone Statistics with the zonestat Utility

To use the zonestat utility, see the zonestat(1) man page and "Using the zonestat Utility in a Non-Global Zone" on page 341.

The zonestat utility reports on the CPU, memory, and resource control utilization of the currently running zones. Each zone's utilization is reported as a percentage of both system resources and the zone's configured limits. The zonestat utility prints a series of reports at specified intervals. Optionally, the utility can print one or more summary reports.

When run from within a non-global zone, only processor sets visible to that zone are reported. The non-global zone output will include all of the memory resources, and the limits resource.

The zonestat service in the global zone must be online to use the zonestat service in the non-global zones. The zonestat service in each non-global zone reads system configuration and utilization data from the zonestat service in the global zone.

The zonestatd system daemon is started during system boot. The daemon monitors the utilization of system resources by zones, as well as zone and system configuration information such as psrset processor sets, pool processor sets, and resource control settings. There are no configurable components.

Non-Global Zone Node Name

The node name in /etc/nodename returned by uname -n can be set by the zone administrator. The node name must be unique.

File Systems and Non-Global Zones

This section provides information about file system issues on an Oracle Solaris system with zones installed. Each zone has its own section of the file system hierarchy, rooted at a directory known as the zone `root`. Processes in the zone can access only files in the part of the hierarchy that is located under the zone root. The `chroot` utility can be used in a zone, but only to restrict the process to a root path within the zone. For more information about `chroot`, see `chroot`(1M).

The `-o nosuid` Option

The `-o nosuid` option to the `mount` utility has the following functionality:

- Processes from a `setuid` binary located on a file system that is mounted using the `nosetuid` option do not run with the privileges of the `setuid` binary. The processes run with the privileges of the user that executes the binary.

  For example, if a user executes a `setuid` binary that is owned by `root`, the processes run with the privileges of the user.
- Opening device-special entries in the file system is not allowed. This behavior is equivalent to specifying the `nodevices` option.

This file system-specific option is available to all Oracle Solaris file systems that can be mounted with `mount` utilities, as described in the `mount`(1M) man page. In this guide, these file systems are listed in "Mounting File Systems in Zones" on page 310. Mounting capabilities are also described. For more information about the `-o nosuid` option, see "Accessing Network File Systems (Reference)" in *System Administration Guide: Network Services*.

Mounting File Systems in Zones

When file systems are mounted from within a zone, the `nodevices` option applies. For example, if a zone is granted access to a block device (`/dev/dsk/c0t0d0s7`) and a raw device (`/dev/rdsk/c0t0d0s7`) corresponding to a UFS file system, the file system is automatically mounted `nodevices` when mounted from within a zone. This rule does not apply to mounts specified through a `zonecfg` configuration.

Options for mounting file systems in non-global zones are described in the following table. Procedures for these mounting alternatives are provided in "Configuring, Verifying, and Committing a Zone" on page 238 and "Mounting File Systems in Running Non-Global Zones" on page 344.

Any file system type not listed in the table can be specified in the configuration if it has a mount binary in `/usr/lib/`*fstype*`/mount`.

Allowing file system mounts other than the default might allow the zone administrator to compromise the system.

File System	Mounting Options in a Non-Global Zone
AutoFS	Cannot be mounted using `zonecfg`. Can be mounted from within the zone.
CacheFS	Cannot be used in a non-global zone.
FDFS	Can be mounted using `zonecfg`, can be mounted from within the zone.
HSFS	Can be mounted using `zonecfg`, can be mounted from within the zone.
LOFS	Can be mounted using `zonecfg`, can be mounted from within the zone.
MNTFS	Cannot be mounted using `zonecfg`. Can be mounted from within the zone.
NFS	Cannot be mounted using `zonecfg`. V2, V3, and V4, which are the versions currently supported in zones, can be mounted from within the zone.
PCFS	Can be mounted using `zonecfg`, can be mounted from within the zone.
PROCFS	Cannot be mounted using `zonecfg`. Can be mounted from within the zone.
TMPFS	Can be mounted using `zonecfg`, can be mounted from within the zone.
UDFS	Can be mounted using `zonecfg`, can be mounted from within the zone.
UFS	Can be mounted using `zonecfg`, can be mounted from within the zone.
ZFS	Can be mounted using the `zonecfg` `dataset` and `fs` resource types.

For more information, see "How to Configure the Zone" on page 238, "Mounting File Systems in Running Non-Global Zones" on page 344, and the `mount(1M)` man page.

Unmounting File Systems in Zones

The ability to unmount a file system will depend on who performed the initial mount. If a file system is specified as part of the zone's configuration using the `zonecfg` command, then the global zone owns this mount and the non-global zone administrator cannot unmount the file system. If the file system is mounted from within the non-global zone, for example, by specifying the mount in the zone's `/etc/vfstab` file, then the non-global zone administrator can unmount the file system.

Security Restrictions and File System Behavior

There are security restrictions on mounting certain file systems from within a zone. Other file systems exhibit special behavior when mounted in a zone. The list of modified file systems follows.

AutoFS

Autofs is a client-side service that automatically mounts the appropriate file system. When a client attempts to access a file system that is not presently mounted, the AutoFS file system intercepts the request and calls `automountd` to mount the requested directory. AutoFS mounts established within a zone are local to that zone. The mounts cannot be accessed from other zones, including the global zone. The mounts are removed when the zone is halted or rebooted. For more information on AutoFS, see "How Autofs Works" in *System Administration Guide: Network Services*.

Each zone runs its own copy of `automountd`. The auto maps and timeouts are controlled by the zone administrator. You cannot trigger a mount in another zone by crossing an AutoFS mount point for a non-global zone from the global zone.

Certain AutoFS mounts are created in the kernel when another mount is triggered. Such mounts cannot be removed by using the regular `umount` interface because they must be mounted or unmounted as a group. Note that this functionality is provided for zone shutdown.

MNTFS

MNTFS is a virtual file system that provides read-only access to the table of mounted file systems for the local system. The set of file systems visible by using `mnttab` from within a non-global zone is the set of file systems mounted in the zone, plus an entry for root (`/`). Mount points with a special device that is not accessible from within the zone, such as `/dev/rdsk/c0t0d0s0`, have their special device set to the same as the mount point. All mounts in the system are visible from the global zone's `/etc/mnttab` table. For more information on MNTFS, see Chapter 20, "Mounting and Unmounting File Systems (Tasks)," in *System Administration Guide: Devices and File Systems*.

NFS

NFS mounts established within a zone are local to that zone. The mounts cannot be accessed from other zones, including the global zone. The mounts are removed when the zone is halted or rebooted.

As documented in the `mount_nfs(1M)` man page, an NFS server should not attempt to mount its own file systems. Thus, a zone should not NFS mount a file system exported by the global zone. From within a zone, NFS mounts behave as though mounted with the `nodevices` option.

The `nfsstat` command output only pertains to the zone in which the command is run. For example, if the command is run in the global zone, only information about the global zone is reported. For more information about the `nfsstat` command, see `nfsstat(1M)`.

The `zlogin` command will fail if any of its open files or any portion of its address space reside on NFS. For more information, see "`zlogin` Command" on page 279.

PROCFS

The `/proc` file system, or PROCFS, provides process visibility and access restrictions as well as information about the zone association of processes. Only processes in the same zone are visible through `/proc`.

Processes in the global zone can observe processes and other objects in non-global zones. This allows such processes to have system-wide observability.

From within a zone, `procfs` mounts behave as though mounted with the `nodevices` option. For more information about `procfs`, see the `proc(4)` man page.

LOFS

The scope of what can be mounted through LOFS is limited to the portion of the file system that is visible to the zone. Hence, there are no restrictions on LOFS mounts in a zone.

UFS, UDFS, PCFS, and other storage-based file systems

When using the `zonecfg` command to configure storage-based file systems that have an `fsck` binary, such as UFS, the zone administrator must specify a `raw` parameter. The parameter indicates the raw (character) device, such as `/dev/rdsk/c0t0d0s7`. `zoneadmd` automatically runs the `fsck` command in non-interactive check-only mode (`fsck -m`) on this device before it mounts the file system. If the `fsck` fails, `zoneadmd` cannot bring the zone to the ready state. The path specified by `raw` cannot be a relative path.

It is an error to specify a device to `fsck` for a file system that does not provide an `fsck` binary in `/usr/lib/fs/`*fstype*`/fsck`. It is also an error if you do not specify a device to `fsck` if an `fsck` binary exists for that file system.

For more information, see "The `zoneadmd` Daemon" on page 257 and the `fsck(1M)`

ZFS

You can add a ZFS dataset to a non-global zone by using the `zonecfg` command with the `add dataset` resource. The dataset will be visible and mounted in the non-global zone and no longer visible in the global zone. The zone administrator can create and destroy file systems within that dataset, and modify the properties of the dataset.

The `zoned` attribute of `zfs` indicates whether a dataset has been added to a non-global zone.

```
# zfs get zoned tank/sales
NAME          PROPERTY    VALUE      SOURCE
tank/sales    zoned       on         local
```

If you want to share a dataset from the global zone, you can add an LOFS-mounted ZFS file system by using the `zonecfg` command with the `add fs` subcommand. The global administrator or a user granted the appropriate authorizations is responsible for setting and controlling the properties of the dataset.

For more information on ZFS, see Chapter 10, "Oracle Solaris ZFS Advanced Topics," in *Oracle Solaris ZFS Administration Guide*.

Non-Global Zones as NFS Clients

Zones can be NFS clients. Version 2, version 3, and version 4 protocols are supported. For information on these NFS versions, see "Features of the NFS Service" in *System Administration Guide: Network Services*. .

The default version is NFS version 4. You can enable other NFS versions on a client by using one of the following methods:

- **Moved to SMF. Use sharectl(1M) to manage Get SMF replacement for /etc/default/nfs**

  You can edit `/etc/default/nfs` to set `NFS_CLIENT_VERSMAX=`*number* so that the zone uses the specified version by default. See "Setting Up NFS Services" in *System Administration Guide: Network Services*. Use the procedure "How to Select Different Versions of NFS on a Client by Modifying the `/etc/default/nfs` File" from the task map.
- You can manually create a version mount. This method overrides the contents of `/etc/default/nfs`. See "Setting Up NFS Services" in *System Administration Guide: Network Services*. Use the procedure How to Use the Command Line to Select Different Versions of NFS on a Client from the task map.

Use of `mknod` Prohibited in a Zone

Note that you cannot use the `mknod` command documented in the `mknod`(1M) man page to make a special file in a non-global zone.

Traversing File Systems

A zone's file system namespace is a subset of the namespace accessible from the global zone. Unprivileged processes in the global zone are prevented from traversing a non-global zone's file system hierarchy through the following means:

- Specifying that the zone root's parent directory is owned, readable, writable, and executable by root only
- Restricting access to directories exported by `/proc`

Get SMF replacement for autofs

Note that attempting to access AutoFS nodes mounted for another zone will fail. The global administrator must not have auto maps that descend into other zones.

Restriction on Accessing A Non-Global Zone From the Global Zone

After a non-global zone is installed, the zone must never be accessed directly from the global zone by any commands other than system backup utilities. Moreover, a non-global zone can no longer be considered secure after it has been exposed to an unknown environment. An example would be a zone placed on a publicly accessible network, where it would be possible for the zone to be compromised and the contents of its file systems altered. If there is any possibility that compromise has occurred, the global administrator should treat the zone as untrusted.

Any command that accepts an alternative root by using the `-R` or `-b` options (or the equivalent) must *not* be used when the following are true:

- The command is run in the global zone.
- The alternative root refers to any path within a non-global zone, whether the path is relative to the current running system's global zone or the global zone in an alternative root.

An example is the `-R` *root_path* option to the `pkgadd` utility run from the global zone with a non-global zone root path.

The list of commands, programs, and utilities that use `-R` with an alternative root path include the following:

- `auditreduce`
- `bart`
- `installf`
- `localeadm`
- `makeuuid`
- `metaroot`
- `pkg`

- `prodreg`
- `removef`
- `routeadm`
- `showrev`
- `syseventadm`

The list of commands and programs that use `-b` with an alternative root path include the following:

- `add_drv`
- `pprosetup`
- `rem_drv`
- `roleadd`
- `sysidconfig`
- `update_drv`
- `useradd`

Networking in Shared-IP Non-Global Zones

On a Solaris system with zones installed, the zones can communicate with each other over the network. The zones all have separate bindings, or connections, and the zones can all run their own server daemons. These daemons can listen on the same port numbers without any conflict. The IP stack resolves conflicts by considering the IP addresses for incoming connections. The IP addresses identify the zone.

Shared-IP Zone Partitioning

The IP stack in a system supporting zones implements the separation of network traffic between zones. Applications that receive IP traffic can only receive traffic sent to the same zone.

Each logical interface on the system belongs to a specific zone, the global zone by default. Logical network interfaces assigned to zones though the `zonecfg` utility are used to communicate over the network. Each stream and connection belongs to the zone of the process that opened it.

Bindings between upper-layer streams and logical interfaces are restricted. A stream can only establish bindings to logical interfaces in the same zone. Likewise, packets from a logical interface can only be passed to upper-layer streams in the same zone as the logical interface.

Each zone has its own set of binds. Each zone can be running the same application listening on the same port number without binds failing because the address is already in use. Each zone can run its own version of the following services:

- Internet services daemon with a full configuration file (see the `inetd(1M)` man page)

- `sendmail` (see the `sendmail`(1M) man page)
- `apache` (see the `apache`(1M) man page)

Zones other than the global zone have restricted access to the network. The standard TCP and UDP socket interfaces are available, but `SOCK_RAW` socket interfaces are restricted to Internet Control Message Protocol (ICMP). ICMP is necessary for detecting and reporting network error conditions or using the `ping` command.

Shared-IP Network Interfaces

Each non-global zone that requires network connectivity has one or more dedicated IP addresses. These addresses are associated with logical network interfaces that can be placed in a zone by using the `ifconfig` command. Zone network interfaces configured by `zonecfg` will automatically be set up and placed in the zone when it is booted. The `ifconfig` command can be used to add or remove logical interfaces when the zone is running. Only the global administrator or a user granted the appropriate authorizations can modify the interface configuration and the network routes.

Within a non-global zone, only that zone's interfaces will be visible to `ifconfig`.

For more information, see the `ifconfig`(1M) and `if_tcp`(7P) man pages.

IP Traffic Between Shared-IP Zones on the Same Machine

Between two zones on the same machine, packet delivery is only allowed if there is a “matching route” for the destination and the zone in the forwarding table.

The matching information is implemented as follows:

- The source address for the packets is selected on the output interface specified by the matching route.
- By default, traffic is permitted between two zones that have addresses on the same subnet. The matching route in this case is the interface route for the subnet.
- If there is a default route for a given zone, where the gateway is on one of the zone's subnets, traffic from that zone to all other zones is allowed. The matching route in this case is the default route.
- If there is a matching route with the `RTF_REJECT` flag, packets trigger an ICMP unreachable message. If there is a matching route with the `RTF_BLACKHOLE` flag, packets are discarded. The global administrator can use the `route` command options described in the following table to create routes with these flags.

Modifier	Flag	Description
`-reject`	`RTF_REJECT`	Emit an ICMP unreachable message when matched.
`-blackhole`	`RTF_BLACKHOLE`	Silently discard packets during updates.

For more information, see the `route`(1M)

Oracle Solaris IP Filter in Shared-IP Zones

Oracle Solaris IP Filter provides stateful packet filtering and network address translation (NAT). A stateful packet filter can monitor the state of active connections and use the information obtained to determine which network packets to allow through the firewall. Oracle Solaris IP Filter also includes stateless packet filtering and the ability to create and manage address pools. See Chapter 24, "IP Filter in Oracle Solaris (Overview)," in *System Administration Guide: IP Services* for additional information.

Oracle Solaris IP Filter can be enabled in non-global zones by turning on loopback filtering as described in Chapter 25, " IP Filter (Tasks)," in *System Administration Guide: IP Services*.

Oracle Solaris IP Filter is derived from open source IP Filter software.

IP Network Multipathing in Shared-IP Zones

IP network multipathing (IPMP) provides physical interface failure detection and transparent network access failover for a system with multiple interfaces on the same IP link. IPMP also provides load spreading of packets for systems with multiple interfaces.

All network configuration is done in the global zone. You can configure IPMP in the global zone, then extend the functionality to non-global zones. The functionality is extended by placing the zone's address in an IPMP group when you configure the zone. Then, if one of the interfaces in the global zone fails, the non-global zone addresses will migrate to another network interface card.

In a given non-global zone, only the interfaces associated with the zone are visible through the `ifconfig` command.

See "How to Extend IP Network Multipathing Functionality to Shared-IP Non-Global Zones" on page 350. The zones configuration procedure is covered in "How to Configure the Zone" on page 238. For information on IPMP features, components, and usage, see Chapter 13, "Introducing IPMP," in *System Administration Guide: Network Interfaces and Network Virtualization*.

Networking in Exclusive-IP Non-Global Zones

An exclusive-IP zone has its own IP-related state and tuning variables. The zone is assigned its own set of data-links when the zone is configured.

For information on features that can be used in an exclusive-IP non-global zone, see "Exclusive-IP Non-Global Zones" on page 211. For information on tuning IP `ndd` variables, see *Oracle Solaris Tunable Parameters Reference Manual*.

Exclusive-IP Zone Partitioning

Exclusive-IP zones have separate TCP/IP stacks, so the separation reaches down to the data-link layer. One or more data-link names, which can be a NIC or a VLAN on a NIC, are assigned to an exclusive-IP zone by the global administrator. The zone administrator can configure IP on those data-links with the same flexibility and options as in the global zone.

Exclusive-IP Data-Link Interfaces

A data-link name must be assigned exclusively to a single zone.

The `dladm show-link` command can be used to display data-links assigned to running zones.

For more information, see `dladm`(1M)

IP Traffic Between Exclusive-IP Zones on the Same Machine

There is no internal loopback of IP packets between exclusive-IP zones. All packets are sent down to the data-link. Typically, this means that the packets are sent out on a network interface. Then, devices like Ethernet switches or IP routers can forward the packets toward their destination, which might be a different zone on the same machine as the sender.

Oracle Solaris IP Filter in Exclusive-IP Zones

You have the same IP Filter functionality that you have in the global zone in an exclusive-IP zone. IP Filter is also configured the same way in exclusive-IP zones and the global zone.

IP Network Multipathing in Exclusive-IP Zones

IP network multipathing (IPMP) provides physical interface failure detection and transparent network access failover for a system with multiple interfaces on the same IP link. IPMP also provides load spreading of packets for systems with multiple interfaces.

The data-link configuration is done in the global zone. First, multiple data-link interfaces are assigned to a zone using `zonecfg`. The multiple data-link interfaces must be attached to the same IP subnet. IPMP can then be configured from within the exclusive-IP zone by the zone administrator.

Device Use in Non-Global Zones

The set of devices available within a zone is restricted to prevent a process in one zone from interfering with processes running in other zones. For example, a process in a zone cannot modify kernel memory or modify the contents of the root disk. Thus, by default, only certain pseudo-devices that are considered safe for use in a zone are available. Additional devices can be made available within specific zones by using the `zonecfg` utility.

`/dev` and the `/devices` Namespace

The `devfs` file system described in the `devfs`(7FS) man page is used by the Oracle Solaris system to manage `/devices`. Each element in this namespace represents the physical path to a hardware device, pseudo-device, or nexus device. The namespace is a reflection of the device tree. As such, the file system is populated by a hierarchy of directories and device special files.

Devices are grouped according to the relative `/dev` hierarchy. For example, all of the devices under `/dev` in the global zone are grouped as global zone devices. For a non-global zone, the devices are grouped in a `/dev` directory under the zone's root path. Each group is a mounted `/dev` file system instance that is mounted under the `/dev` directory. Thus, the global zone devices are mounted under `/dev`, while the devices for a non-global zone named `my-zone` are mounted under `/my-zone_rootpath/dev`.

The `/dev` file hierarchy is managed by the `dev` file system described in the `dev`(7FS) man page.

Caution – Subsystems that rely on `/devices` path names are not able to run in non-global zones. The subsystems must be updated to use `/dev` path names.

Exclusive-Use Devices

You might have devices that you want to assign to specific zones. Allowing unprivileged users to access block devices could permit those devices to be used to cause system panic, bus resets, or other adverse effects. Before making such assignments, consider the following issues:

- Before assigning a SCSI tape device to a specific zone, consult the `sgen`(7D) man page.
- Placing a physical device into more than one zone can create a covert channel between zones. Global zone applications that use such a device risk the possibility of compromised data or data corruption by a non-global zone.

Device Driver Administration

In a non-global zone, you can use the modinfo command described in the modinfo(1M) man page to examine the list of loaded kernel modules.

Most operations concerning kernel, device, and platform management will not work inside a non-global zone because modifying platform hardware configurations violates the zone security model. These operations include the following:

- Adding and removing drivers
- Explicitly loading and unloading kernel modules
- Initiating dynamic reconfiguration (DR) operations
- Using facilities that affect the state of the physical platform

Utilities That Do Not Work or Are Modified in Non-Global Zones

Utilities That Do Not Work in Non-Global Zones

The following utilities do not work in a zone because they rely on devices that are not normally available:

- cdrecord (See the man page in the /usr/share/man/man1 directory.)
- cdrw (see the cdrw(1) man page)
- rmformat (see the rmformat(1) man page)
- add_drv (see the add_drv(1M) man page)
- disks (see the disks(1M) man page)
- prtconf (see the prtconf(1M) man page)
- prtdiag (see the prtdiag(1M) man page)
- rem_drv (see the rem_drv(1M) man page)

SPARC: Utility Modified for Use in a Non-Global Zone

The eeprom utility can be used in a zone to view settings. The utility cannot be used to change settings. For more information, see the eeprom(1M) and openprom(7D) man pages.

Running Applications in Non-Global Zones

In general, all applications can run in a non-global zone. However, the following types of applications might not be suitable for this environment:

- Applications that use privileged operations that affect the system as a whole. Examples include operations that set the global system clock or lock down physical memory.
- The few applications dependent upon certain devices that do not exist in a non-global zone, such as `/dev/kmem`.
- In a shared-IP zone, applications dependent upon devices in `/dev/ip`.

Resource Controls Used in Non-Global Zones

For additional information about using a resource management feature in a zone, also refer to the chapter that describes the feature in Part I, "Oracle Solaris Resource Management."

Any of the resource controls and attributes described in the resource management chapters can be set in the global and non-global zone `/etc/project` file, NIS map, or LDAP directory service. The settings for a given zone affect only that zone. A project running autonomously in different zones can have controls set individually in each zone. For example, Project A in the global zone can be set `project.cpu-shares=10` while Project A in a non-global zone can be set `project.cpu-shares=5`. You could have several instances of `rcapd` running on the system, with each instance operating only on its zone.

The resource controls and attributes used in a zone to control projects, tasks, and processes within that zone are subject to the additional requirements regarding pools and the zone-wide resource controls.

A non-global zone can be associated with one resource pool, although the pool need not be exclusively assigned to a particular zone. Multiple non-global zones can share the resources of one pool. Processes in the global zone, however, can be bound by a sufficiently privileged process to any pool. The resource controller `poold` only runs in the global zone, where there is more than one pool for it to operate on. The `poolstat` utility run in a non-global zone displays only information about the pool associated with the zone. The `pooladm` command run without arguments in a non-global zone displays only information about the pool associated with the zone.

Zone-wide resource controls do not take effect when they are set in the `project` file. A zone-wide resource control is set through the `zonecfg` utility.

Fair Share Scheduler on a System With Zones Installed

This section describes how to use the fair share scheduler (FSS) with zones.

FSS Share Division in a Global or Non-Global Zone

FSS CPU shares for a zone are hierarchical. The shares for the global and non-global zones are set by the global administrator through the zone-wide resource control `zone.cpu-shares`. The `project.cpu-shares` resource control can then be defined for each project within that zone to further subdivide the shares set through the zone-wide control.

To assign zone shares by using the `zonecfg` command, see "How to Set `zone.cpu-shares` in the Global Zone" on page 248. For more information on `project.cpu-shares`, see "Available Resource Controls" on page 80. Also see "Using the Fair Share Scheduler on an Oracle Solaris System With Zones Installed" on page 353 for example procedures that show how to set shares on a temporary basis.

Share Balance Between Zones

You can use `zone.cpu-shares` to assign FSS shares in the global zone and in non-global zones. If FSS is the default scheduler on your system and shares are not assigned, each zone is given one share by default. If you have one non-global zone on your system and you give this zone two shares through `zone.cpu-shares`, that defines the proportion of CPU which the non-global zone will receive in relation to the global zone. The ratio of CPU between the two zones is 2:1.

Extended Accounting on a System With Zones Installed

The extended accounting subsystem collects and reports information for the entire system (including non-global zones) when run in the global zone. The global administrator can also determine resource consumption on a per-zone basis.

The extended accounting subsystem permits different accounting settings and files on a per-zone basis for process-based and task-based accounting. The `exacct` records can be tagged with the zone name `EXD PROC ZONENAME` for processes, and the zone name `EXD TASK ZONENAME` for tasks. Accounting records are written to the global zone's accounting files as well as the per-zone accounting files. The `EXD TASK HOSTNAME`, `EXD PROC HOSTNAME`, and `EXD HOSTNAME` records contain the `uname -n` value for the zone in which the process or task executed instead of the global zone's node name.

For information about IPQoS flow accounting, see Chapter 34, "Using Flow Accounting and Statistics Gathering (Tasks)," in *System Administration Guide: IP Services*.

Privileges in a Non-Global Zone

Processes are restricted to a subset of privileges. Privilege restriction prevents a zone from performing operations that might affect other zones. The set of privileges limits the capabilities of privileged users within the zone. To display the list of privileges available from within a given zone, use the `ppriv` utility.

The following table lists all of the Oracle Solaris privileges and the status of each privilege with respect to zones. Optional privileges are not part of the default set of privileges but can be specified through the `limitpriv` property. Required privileges must be included in the resulting privilege set. Prohibited privileges cannot be included in the resulting privilege set.

TABLE 24–1 Status of Privileges in Zones

Privilege	Status	Notes
`cpc_cpu`	Optional	Access to certain `cpc`(3CPC) counters
`dtrace_proc`	Optional	`fasttrap` and `pid` providers; `plockstat`(1M)
`dtrace_user`	Optional	`profile` and `syscall` providers
`graphics_access`	Optional	`ioctl`(2) access to `agpgart_io`(7I)
`graphics_map`	Optional	`mmap`(2) access to `agpgart_io`(7I)
`net_rawaccess`	Optional in shared-IP zones. Default in exclusive-IP zones.	Raw `PF_INET`/`PF_INET6` packet access
`proc_clock_highres`	Optional	Use of high resolution timers
`proc_priocntl`	Optional	Scheduling control; `priocntl`(1)
`sys_ipc_config`	Optional	Raising IPC message queue buffer size
`sys_time`	Optional	System time manipulation; `xntp`(1M)
`dtrace_kernel`	Prohibited	Currently unsupported
`proc_zone`	Prohibited	Currently unsupported
`sys_config`	Prohibited	Currently unsupported
`sys_devices`	Prohibited	Currently unsupported
`sys_linkdir`	Prohibited	Currently unsupported
`sys_net_config`	Prohibited	Currently unsupported
`sys_res_config`	Prohibited	Currently unsupported
`sys_suser_compat`	Prohibited	Currently unsupported

TABLE 24–1 Status of Privileges in Zones *(Continued)*

Privilege	Status	Notes
proc_exec	Required, Default	Used to start init(1M)
proc_fork	Required, Default	Used to start init(1M)
sys_mount	Required, Default	Needed to mount required file systems
sys_ip_config	Required, Default in exclusive-IP zones Prohibited in shared-IP zones	Required to boot zone and initialize IP networking in exclusive-IP zone
contract_event	Default	Used by contract file system
contract_observer	Default	Contract observation regardless of UID
file_chown	Default	File ownership changes
file_chown_self	Default	Owner/group changes for own files
file_dac_execute	Default	Execute access regardless of mode/ACL
file_dac_read	Default	Read access regardless of mode/ACL
file_dac_search	Default	Search access regardless of mode/ACL
file_dac_write	Default	Write access regardless of mode/ACL
file_link_any	Default	Link access regardless of owner
file_owner	Default	Other access regardless of owner
file_setid	Default	Permission changes for setid, setgid, setuid files
ipc_dac_read	Default	IPC read access regardless of mode
ipc_dac_owner	Default	IPC write access regardless of mode
ipc_owner	Default	IPC other access regardless of mode
net_icmpaccess	Default	ICMP packet access: ping(1M)
net_privaddr	Default	Binding to privileged ports
proc_audit	Default	Generation of audit records
proc_chroot	Default	Changing of root directory
proc_info	Default	Process examination

TABLE 24–1 Status of Privileges in Zones *(Continued)*

Privilege	Status	Notes
proc_lock_memory	Default	Locking memory; shmctl(2)and mlock(3C) If this privilege is assigned to a non-global zone by the system administrator, consider also setting the zone.max-locked-memory resource control to prevent the zone from locking all memory.
proc_owner	Default	Process control regardless of owner
proc_session	Default	Process control regardless of session
proc_setid	Default	Setting of user/group IDs at will
proc_taskid	Default	Assigning of task IDs to caller
sys_acct	Default	Management of accounting
sys_admin	Default	Simple system administration tasks
sys_audit	Default	Management of auditing
sys_nfs	Default	NFS client support
sys_resource	Default	Resource limit manipulation
sys_share	Default	Needed to share file systems

The following table lists all of the Oracle Solaris Trusted Extensions privileges and the status of each privilege with respect to zones. Optional privileges are not part of the default set of privileges but can be specified through the limitpriv property.

Note – Oracle Trusted Solaris privileges are interpreted only if the system is configured with Oracle Trusted Extensions.

TABLE 24–2 Status of Oracle Solaris Trusted Extensions Privileges in Zones

Oracle Solaris Trusted Extensions Privilege	Status	Notes
file_downgrade_sl	Optional	Set the sensitivity label of file or directory to a sensitivity label that does not dominate the existing sensitivity label
file_upgrade_sl	Optional	Set the sensitivity label of file or directory to a sensitivity label that dominates the existing sensitivity label

TABLE 24–2 Status of Oracle Solaris Trusted Extensions Privileges in Zones *(Continued)*

Oracle Solaris Trusted Extensions Privilege	Status	Notes
`sys_trans_label`	Optional	Translate labels not dominated by sensitivity label
`win_colormap`	Optional	Colormap restrictions override
`win_config`	Optional	Configure or destroy resources that are permanently retained by the X server
`win_dac_read`	Optional	Read from window resource not owned by client's user ID
`win_dac_write`	Optional	Write to or create window resource not owned by client's user ID
`win_devices`	Optional	Perform operations on input devices.
`win_dga`	Optional	Use direct graphics access X protocol extensions; frame buffer privileges needed
`win_downgrade_sl`	Optional	Change sensitivity label of window resource to new label dominated by existing label
`win_fontpath`	Optional	Add an additional font path
`win_mac_read`	Optional	Read from window resource with a label that dominates the client's label
`win_mac_write`	Optional	Write to window resource with a label not equal to the client's label
`win_selection`	Optional	Request data moves without confirmer intervention
`win_upgrade_sl`	Optional	Change sensitivity label of window resource to a new label not dominated by existing label
`net_bindmlp`	Default	Allows binding to a multilevel port (MLP)
`net_mac_aware`	Default	Allows reading down through NFS

To alter privileges in a non-global zone configuration, see "Configuring, Verifying, and Committing a Zone" on page 238

To inspect privilege sets, see "Using the `ppriv` Utility" on page 339. For more information about privileges, see the `ppriv(1)` man page and *System Administration Guide: Security Services*.

Using IP Security Architecture in Zones

The Internet Protocol Security Architecture (IPsec), which provides IP datagram protection, is described in Chapter 18, "IP Security Architecture (Overview)," in *System Administration Guide: IP Services*. The Internet Key Exchange (IKE) protocol is used to manage the required keying material for authentication and encryption automatically.

For more information, see the `ipsecconf(1M)` and `ipseckey(1M)` man pages.

IP Security Architecture in Shared-IP Zones

IPsec can be used in the global zone. However, IPsec in a non-global zone cannot use IKE. Therefore, you must manage the IPsec keys and policy for the non-global zones by using the Internet Key Exchange (IKE) protocol in the global zone. Use the source address that corresponds to the non-global zone that you are configuring.

IP Security Architecture in Exclusive-IP Zones

IPsec can be used in exclusive-IP zones.

Using Oracle Solaris Auditing in Zones

An audit record describes an event, such as logging in to a system or writing to a file. Oracle Solaris Auditing provides the following two auditing models on systems that are running zones:

- All zones are audited identically from the global zone. This model is used when all zones are administered by the global zone, for example, to achieve service isolation through zones.
- Each zone is audited independently of the global zone. This model is used when each zone is administered separately, for example, to achieve server consolidation by zone.

Oracle Solaris Auditing is described in Chapter 28, "Oracle Solaris Auditing (Overview)," in *System Administration Guide: Security Services*. For zones considerations associated with auditing, see "Auditing on a System With Zones" in *System Administration Guide: Security Services* and "Configuring the Audit Service in Zones (Tasks)" in *System Administration Guide: Security Services*. For additional information, also see the `auditconfig(1M)`, `auditreduce(1M)`, `usermod(1M)`, and `user_attr(4)` man pages.

Note – It is also possible to use audit policies that are activated on a temporary basis, but not set in the repository.

For additional information, see the example that follows "How to Change Audit Policy" in *System Administration Guide: Security Services*.

Core Files in Zones

The `coreadm` command is used to specify the name and location of core files produced by abnormally terminating processes. Core file paths that include the *zonename* of the zone in which the process executed can be produced by specifying the `%z` variable. The path name is relative to a zone's root directory.

For more information, see the `coreadm(1M)` and `core(4)` man pages.

Running DTrace in a Non-Global Zone

DTrace programs that only require the `dtrace_proc` and `dtrace_user` privileges can be run in a non-global zone. To add these privileges to the set of privileges available in the non-global zone, use the `zonecfg` `limitpriv` property. For instructions, see "How to Use DTrace" on page 343.

The providers supported through `dtrace_proc` are `fasttrap` and `pid`. The providers supported through `dtrace_user` are `profile` and `syscall`. DTrace providers and actions are limited in scope to the zone.

Also see "Privileges in a Non-Global Zone" on page 324 for more information.

About Backing Up an Oracle Solaris System With Zones Installed

You can perform backups in individual non-global zones, or back up the entire system from the global zone.

Backing Up Loopback File System Directories

Do not back up the loopback file systems (`lofs`) in non-global zones. An attempt by the zone administrator to restore `lofs` file systems from a non-global zone could cause a serious problem.

Backing Up Your System From the Global Zone

You might choose to perform your backups from the global zone in the following cases:

- You want to back up the configurations of your non-global zones as well as the application data.
- Your primary concern is the ability to recover from a disaster. If you need to restore everything or almost everything on your system, including the root file systems of your zones and their configuration data as well as the data in your global zone, backups should take place in the global zone.
- You have commercial network backup software.

Note – Your network backup software should be configured to skip all inherited `lofs` file systems if possible. The backup should be performed when the zone and its applications have quiesced the data to be backed up.

Backing Up Individual Non-Global Zones on Your System

You might decide to perform backups within the non-global zones in the following cases.

- The non-global zone administrator needs the ability to recover from less serious failures or to restore application or user data specific to a zone.
- You want to use programs that back up on a file-by-file basis, such as `tar` or `cpio`. See the `tar(1)` and `cpio(1)` man pages.
- You use the backup software of a particular application or service running in a zone. It might be difficult to execute the backup software from the global zone because application environments, such as directory path and installed software, would be different between the global zone and the non-global zone.

  If the application can perform a snapshot on its own backup schedule in each non-global zone and store those backups in a writable directory exported from the global zone, the global zone administrator can pick up those individual backups as part of the backup strategy from the global zone.

Creating Oracle Solaris ZFS Backups

The ZFS `send` command creates a stream representation of a ZFS snapshot that is written to standard output. By default, a full stream is generated. You can redirect the output to a file or to a different system. The ZFS `receive` command creates a snapshot in which contents are specified in the stream that is provided on standard input. If a full stream is received, a new file system is created as well. You can send ZFS snapshot data and receive ZFS snapshot data and file systems with these commands.

In addition to the ZFS `send` and `receive` commands, you can also use archive utilities, such as the `tar` and `cpio` commands, to save ZFS files. These utilities save and restore ZFS file attributes and access control lists (ACLs). Check the appropriate options for both the `tar` and `cpio` commands in the man pages.

For information and examples, see Chapter 7, "Working With Oracle Solaris ZFS Snapshots and Clones," in *Oracle Solaris ZFS Administration Guide*.

Determining What to Back Up in Non-Global Zones

You can back up everything in the non-global zone, or, because a zone's configuration changes less frequently, you can perform backups of the application data only.

Backing Up Application Data Only

If application data is kept in a particular part of the file system, you might decide to perform regular backups of this data only. The zone's root file system might not have to be backed up as often because it changes less frequently.

You will have to determine where the application places its files. Locations where files can be stored include the following:

- Users' home directories
- `/etc` for configuration data files
- `/var`

Assuming the application administrator knows where the data is stored, it might be possible to create a system in which a per-zone writable directory is made available to each zone. Each zone can then store its own backups, and the global administrator or user granted the appropriate authorizations can make this location one of the places on the system to back up.

General Database Backup Operations

If the database application data is not under its own directory, the following rules apply:

- Ensure that the databases are in a consistent state first.

  Databases must be quiesced because they have internal buffers to flush to disk. Make sure that the databases in non-global zones have come down before starting the backup from the global zone.

- Within each zone, use file system features to make a snapshot of the data, then back up the snapshots directly from the global zone.

  This process will minimize elapsed time for the backup window and remove the need for backup clients/modules in all of the zones.

Tape Backups

Each non-global zone can take a snapshot of its private file systems when it is convenient for that zone and the application has been briefly quiesced. Later, the global zone can back up each of the snapshots and put them on tape after the application is back in service.

This method has the following advantages:

- Fewer tape devices are needed.
- There is no need for coordination between the non-global zones.
- There is no need to assign devices directly to zones, which improves security.
- Generally, this method keeps system management in the global zone, which is preferred.

About Restoring Non-Global Zones

In the case of a restore where the backups were done from the global zone, the global administrator or a user granted the appropriate authorizations can reinstall the affected zones and then restore that zone's files. Note that this assumes the following:

- The zone being restored has the same configuration as it did when the backup was done.
- The global zone has not been updated between the time when the backup was done and the time when the zone is restored.

Otherwise, the restore could overwrite some files that should be merged by hand.

Note – If all file systems in the global zone are lost, restoring everything in the global zone restores the non-global zones as well, as long as the respective root file systems of the non-global zones were included in the backup.

Commands Used on a System With Zones Installed

The commands identified in Table 24–3 provide the primary administrative interface to the zones facility.

TABLE 24–3 Commands Used to Administer Zones

Command Reference	Description
zlogin(1)	Log in to a non-global zone
zonename(1)	Prints the name of the current zone
zoneadm(1M)	Administers zones on a system
zonecfg(1M)	Used to set up a zone configuration
getzoneid(3C)	Used to map between zone ID and name
zones(5)	Provides description of zones facility
zcons(7D)	Zone console device driver

The `zoneadmd` daemon is the primary process for managing the zone's virtual platform. The man page for the `zoneadmd` daemon is `zoneadmd`(1M). The daemon does not constitute a programming interface.

The commands in the next table are used with the resource capping daemon.

TABLE 24–4 Commands Used With `rcapd`

Command Reference	Description
rcapstat(1)	Monitors the resource utilization of capped projects.
rcapadm(1M)	Configures the resource capping daemon, displays the current status of the resource capping daemon if it has been configured, and enables or disables resource capping
rcapd(1M)	The resource capping daemon.

The commands identified in the following table have been modified for use on an Oracle Solaris system with zones installed. These commands have options that are specific to zones or present information differently. The commands are listed by man page section.

TABLE 24–5 Commands Modified for Use on an Oracle Solaris System With Zones Installed

Command Reference	Description
ipcrm(1)	Added -z *zone* option. This option is only useful when the command is executed in the global zone.
ipcs(1)	Added -z *zone* option. This option is only useful when the command is executed in the global zone.
pgrep(1)	Added -z *zoneidlist* option. This option is only useful when the command is executed in the global zone.
ppriv(1)	Added the expression zone for use with the -l option to list all privileges available in the current zone. Also use the option -v after zone to obtain verbose output.
priocntl(1)	Zone ID can be used in *idlist* and -i *idtype* to specify processes. You can use the priocntl -i *zoneid* command to move running processes into a different scheduling class in a non-global zone.
proc(1)	Added -z *zone* option to ptree only. This option is only useful when the command is executed in the global zone.
ps(1)	Added *zonename* and *zoneid* to list of recognized format names used with the -o option. Added -z *zonelist* to list only processes in the specified zones. Zones can be specified either by zone name or by zone ID. This option is only useful when the command is executed in the global zone. Added -Z to print the name of the zone associated with the process. The name is printed under an additional column header, ZONE.
renice(1)	Added *zoneid* to list of valid arguments used with the -i option.
sar(1)	If executed in a non-global zone in which the pools facility is enabled, the -b, -c -g, -m, -p, -u, -w, and -y options display values only for processors that are in the processor set of the pool to which the zone is bound.
auditconfig(1M)	Added zonename token.
auditreduce(1M)	Added -z *zone-name* option. Added ability to get an audit log of a zone.
coreadm(1M)	Added variable %z to identify the zone in which process executed.
df(1M)	Added -Z option to display mounts in all visible zones. This option has no effect in a non-global zone.
ifconfig(1M)	Added zone option for global zone use (the default), and -zone *zonename* for non-global zone use.
iostat(1M)	If executed in a non-global zone in which the pools facility is enabled, information is provided only for those processors that are in the processor set of the pool to which the zone is bound.

TABLE 24–5 Commands Modified for Use on an Oracle Solaris System With Zones Installed *(Continued)*

Command Reference	Description
kstat(1M)	If executed in the global zone, kstats are displayed for all zones. If executed in a non-global zone, only kstats with a matching *zoneid* are displayed.
mpstat(1M)	If executed in a non-global zone in which the pools facility is enabled, command only displays lines for the processors that are in the processor set of the pool to which the zone is bound.
ndd(1M)	When used in the global zone, displays information for all zones. ndd on the TCP/IP modules in an exclusive-IP zone only displays information for that zone.
netstat(1M)	Displays information for the current zone only.
nfsstat(1M)	Displays statistics for the current zone only.
poolbind(1M)	Added *zoneid* list. Also see "Resource Pools Used in Zones" on page 137 for information about using zones with resource pools.
prstat(1M)	Added -z *zoneidlist* option. Also added -Z option. If executed in a non-global zone in which the pools facility is enabled, the percentage of recent CPU time used by the process is displayed only for the processors in the processor set of the pool to which the zone is bound. Output of the -a, -t, -T, -J, and -Z options displays a SWAP instead of a SIZE column. The swap reported is the total swap consumed by the zone's processes and tmpfs mounts. This value assists in monitoring the swap reserved by each zone, which can be used to choose a reasonable zone.max-swap setting.
psrinfo(1M)	If executed in a non-global zone, only information about the processors visible to the zone is displayed.
traceroute(1M)	Usage change. When specified from within a non-global zone, the -F option has no effect because the "don't fragment" bit is always set.
vmstat(1M)	When executed in a non-global zone in which the pools facility is enabled, statistics are reported only for the processors in the processor set of the pool to which the zone is bound. Applies to output from the -p option and the page, faults, and cpu report fields.
auditon(2)	Added AUDIT_ZONENAME to generate a zone ID token with each audit record.
priocntl(2)	Added P_ZONEID *id* argument.
processor_info(2)	If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.

TABLE 24–5 Commands Modified for Use on an Oracle Solaris System With Zones Installed *(Continued)*

Command Reference	Description
p_online(2)	If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.
pset_bind(2)	Added P_ZONEID as *idtype*. Added zone to possible choices for P_MYID specification. Added P_ZONEID to valid *idtype* list in EINVAL error description.
pset_info(2)	If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.
pset_list(2)	If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.
pset_setattr(2)	If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.
sysinfo(2)	Changed PRIV_SYS_CONFIG to PRIV_SYS_ADMIN.
umount(2)	ENOENT is returned if file pointed to by *file* is not an absolute path.
getloadavg(3C)	If the caller is in a non-global zone and the pools facility is enabled, the behavior is equivalent to calling with a psetid of PS_MYID.
getpriority(3C)	Added zone IDs to target processes that can be specified. Added zone ID to EINVAL error description.
priv_str_to_set(3C)	Added "zone" string for the set of all privileges available within the caller's zone.
pset_getloadavg(3C)	If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.
sysconf(3C)	If the caller is in a non-global zone and the pools facility enabled, sysconf(_SC_NPROCESSORS_CONF) and sysconf(_SC_NPROCESSORS_ONLN) return the number of total and online processors in the processor set of the pool to which the zone is bound.
ucred_get(3C)	Added ucred_getzoneid() function, which returns the zone ID of the process or -1 if the zone ID is not available.
core(4)	Added n_type: NT_ZONENAME. This entry contains a string that describes the name of the zone in which the process was running.
pkginfo(4)	Now provides optional parameters and an environment variable in support of zones.

TABLE 24–5 Commands Modified for Use on an Oracle Solaris System With Zones Installed *(Continued)*

Command Reference	Description
proc(4)	Added capability to obtain information on processes running in zones.
audit_syslog(5)	Added in<*zone name*> field that is used if the zonename audit policy is set.
privileges(5)	Added PRIV_PROC_ZONE, which allows a process to trace or send signals to processes in other zones. See zones(5).
if_tcp(7P)	Added zone ioctl() calls.
cmn_err(9F)	Added zone parameter.
ddi_cred(9F)	Added crgetzoneid(), which returns the zone ID from the user credential pointed to by cr.

CHAPTER 25

Administering Oracle Solaris Zones (Tasks)

This chapter covers general administration tasks and provides usage examples.

See Chapter 24, "Oracle Solaris Zones Administration (Overview)," for general zone administration topics.

Using the `ppriv` Utility

Use the `ppriv` utility to display the zone's privileges.

▼ How to List Oracle Solaris Privileges in the Global Zone

Use the `ppriv` utility with the `-l` option to list the privileges available on the system.

● **At the prompt, type `ppriv -l zone` to report the set of privileges available in the zone.**

```
global# ppriv -l zone
```

You will see a display similar to this:

```
contract_event
contract_observer
cpc_cpu
.
.
.
```

How to List the Non-Global Zone's Privilege Set

Use the `ppriv` utility with the `-l` option and the expression `zone` to list the zone's privileges.

1. **Log into the non-global zone. This example uses a zone named *my-zone*.**
2. **At the prompt, type `ppriv -l zone` to report the set of privileges available in the zone.**

```
my-zone# ppriv -l zone
```

You will see a display similar to this:

```
contract_event
contract_observer
file_chown

.
.
.
```

How to List a Non-Global Zone's Privilege Set With Verbose Output

Use the `ppriv` utility with the `-l` option, the expression `zone`, and the `-v` option to list the zone's privileges.

1. **Log into the non-global zone. This example uses a zone named *my-zone*.**
2. **At the prompt, type `ppriv -l -v zone` to report the set of privileges available in the zone, with a description of each privilege.**

```
my-zone# ppriv -lv zone
```

You will see a display similar to this:

```
contract_event
        Allows a process to request critical events without limitation.
        Allows a process to request reliable delivery of all events on
        any event queue.
contract_observer
```

```
        Allows a process to observe contract events generated by
        contracts created and owned by users other than the process's
        effective user ID.
        Allows a process to open contract event endpoints belonging to
        contracts created and owned by users other than the process's
        effective user ID.
file_chown
        Allows a process to change a file's owner user ID.
        Allows a process to change a file's group ID to one other than
        the process' effective group ID or one of the process'
        supplemental group IDs.
.
.
.
```

Using the `zonestat` Utility in a Non-Global Zone

The `zonestat` utility reports on the CPU, memory, and resource control utilization of the currently running zones. Two usage examples follow. For more information, see `zonestat(1)`.

▼ How to Use the `zonestat` Utility to Display a Summary of CPU and Memory Utilization

1 **Be superuser, or have equivalent authorizations.**

2 **Display a summary of CPU and memory utilization every 5 seconds.**

```
# zonestat -z global -r physical-memory 5
# zonestat 5 1
Collecting data for first interval...
Interval: 1, Duration: 0:00:05
SUMMARY                    Cpus/Online: 2/2   Physical: 2046M    Virtual: 2909M
                    ----------CPU---------- ----PHYSICAL----- -----VIRTUAL-----
               ZONE  USED %PART  %CAP %SHRU  USED   PCT  %CAP  USED   PCT  %CAP
            [total]  0.01 0.99%     -     -  693M 33.8%     -  967M 33.2%     -
           [system]  0.00 0.15%     -     -  336M 16.4%     -  682M 23.4%     -
             global  0.01 0.80%     -     -  276M 13.4%     -  217M 7.49%     -
              zone1  0.00 0.01%     -     - 40.2M 1.96%     - 33.1M 1.13%     -
              zone2  0.00 0.01%     -     - 40.2M 1.96%     - 33.4M 1.14%     -
```

▼ How to Use the `zonestat` Utility to Report on the Default `pset`

1 **Be superuser, or have equivalent authorizations.**

2 **Report on the default `pset` once a second for 1 minute:**

```
# zonestat -r default-pset 1 1m
Collecting data for first interval...
Interval: 1, Duration: 0:00:01
PROCESSOR_SET                     TYPE  ONLINE/CPUS     MIN/MAX
pset_default              default-pset          2/2         1/-
                                  ZONE  USED   PCT   CAP  %CAP   SHRS  %SHR %SHRU
                               [total]  0.02 1.10%     -     -      -     -     -
                              [system]  0.00 0.19%     -     -      -     -     -
                                global  0.01 0.77%     -     -      -     -     -
                                 zone1  0.00 0.07%     -     -      -     -     -
                                 zone2  0.00 0.06%     -     -      -     -     -

...
Interval: 60, Duration: 0:01:00
PROCESSOR_SET                     TYPE  ONLINE/CPUS     MIN/MAX
pset_default              default-pset          2/2         1/-
                                  ZONE  USED   PCT   CAP  %CAP   SHRS  %SHR %SHRU
                               [total]  0.06 3.26%     -     -      -     -     -
                              [system]  0.00 0.18%     -     -      -     -     -
                                global  0.05 2.94%     -     -      -     -     -
                                 zone1  0.00 0.06%     -     -      -     -     -
                                 zone2  0.00 0.06%     -     -      -     -     -
```

▼ Using `zonestat` to Report Total and High Utilization

1 **Be superuser, or have equivalent authorizations.**

2 **Monitor silently at a 10–second interval for 1 hour, then produce a report on the total and high utilizations.**

```
# zonestat -q -R total,high 10s 1h 1h
Report: Total Usage
    Start: Wed Nov  3 06:55:20 PDT 2010
      End: Wed Nov  3 07:46:30 PDT 2010
    Intervals: 308, Duration: 0:51:10
SUMMARY                    Cpus/Online: 2/2   Physical: 2046M    Virtual: 2909M
                    ----------CPU---------- ----PHYSICAL----- -----VIRTUAL-----
               ZONE  USED %PART  %CAP %SHRU  USED   PCT  %CAP  USED   PCT  %CAP
            [total]  0.01 0.66%     -     -  693M 33.8%     -  967M 33.2%     -
           [system]  0.00 0.13%     -     -  336M 16.4%     -  682M 23.4%     -
             global  0.01 0.51%     -     -  276M 13.4%     -  218M 7.49%     -
              zone1  0.00 0.01%     -     - 40.2M 1.96%     - 33.1M 1.13%     -
              zone2  0.00 0.01%     -     - 40.2M 1.96%     - 33.4M 1.14%     -

Report: High Usage
```

```
    Start: Wed Nov  3 06:55:20 PDT 2010
      End: Wed Nov  3 07:46:30 PDT 2010
    Intervals: 308, Duration: 0:51:10
SUMMARY                   Cpus/Online: 2/2   Physical: 2046M    Virtual: 2909M
                    ----------CPU---------- ----PHYSICAL----- -----VIRTUAL-----
               ZONE  USED %PART  %CAP %SHRU  USED   PCT  %CAP  USED   PCT  %CAP
            [total]  0.01 0.82%     -     -  693M 33.8%     -  967M 33.2%     -
           [system]  0.00 0.19%     -     -  336M 16.4%     -  682M 23.4%     -
             global  0.01 0.59%     -     -  276M 13.4%     -  217M 7.49%     -
              zone1  0.00 0.01%     -     - 40.2M 1.96%     - 33.1M 1.13%     -
              zone2  0.00 0.01%     -     - 40.2M 1.96%     - 33.4M 1.14%     -
```

Using DTrace in a Non-Global Zone

Perform the following steps to use DTrace functionality as described in "Running DTrace in a Non-Global Zone" on page 329.

How to Use DTrace

1. **Use the `zonecfg limitpriv` property to add the `dtrace_proc` and `dtrace_user` privileges.**

```
global# zonecfg -z my-zone
zonecfg:my-zone> set limitpriv="default,dtrace_proc,dtrace_user"
zonecfg:my-zone> exit
```

Note – Depending on your requirements, you can add either privilege, or both privileges.

---

2. **Boot the zone.**

```
global# zoneadm -z my-zone boot
```

3. **Log in to the zone.**

```
global# zlogin my-zone
```

4. **Run the DTrace program.**

```
my-zone# dtrace -l
```

Checking the Status of SMF Services in a Non-Global Zone

To check the status of SMF services in a non-global zone, use the `zlogin` command.

▼ How to Check the Status of SMF Services From the Command Line

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **From the command line, type the following to show all services, including disabled ones.**

   ```
   global# zlogin my-zone svcs -a
   ```

See Also For more information, see Chapter 21, "Logging In to Non-Global Zones (Tasks)," and svcs(1).

▼ How to Check the Status of SMF Services From Within a Zone

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Log in to the zone.**

   ```
   global# zlogin my-zone
   ```

3. **Run the `svcs` command with the `-a` option to show all services, including disabled ones.**

   ```
   my-zone# svcs -a
   ```

See Also For more information, see Chapter 21, "Logging In to Non-Global Zones (Tasks)," and svcs(1).

Mounting File Systems in Running Non-Global Zones

You can mount file systems in a running non-global zone. The following procedures are covered.

- As the global administrator or a user granted the appropriate authorizations in the global zone, you can import raw and block devices into a non-global zone. After the devices are imported, the zone administrator has access to the disk. The zone administrator can then create a new file system on the disk and perform one of the following actions:
  - Mount the file system manually
  - Place the file system in `/etc/vfstab` so that it will be mounted on zone boot

- As the global administratoror a user granted the appropriate authorizations , you can also mount a file system from the global zone into the non-global zone.

  Before mounting a file system from the global zone into a non-global zone, note that the non-global zone should be in the ready state or be booted. Otherwise, the next attempt to ready or boot the zone will fail. In addition, any file systems mounted from the global zone into a non-global zone will be unmounted when the zone halts.

Tip – If proposed support of zones with multiple boot environments is available in a future release, global administrators should not mount global zone file systems into a non-global zone. This action would break the ability to switch between boot environments.

▼ How to Use LOFS to Mount a File System

You can share a file system between the global zone and non-global zones by using LOFS mounts. This procedure uses the `zonecfg` command to add an LOFS mount of the global zone `/export/datafiles` file system to the `my-zone` configuration. This example does not customize the mount options.

You must be the global administrator or a user in the global zone with the Zone Security rights profile to perform this procedure.

1. **Be superuser, or have the required rights profile.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Use the `zonecfg` command.**

   ```
   global# zonecfg -z my-zone
   ```

3. **Add a file system to the configuration.**

   ```
   zonecfg:my-zone> add fs
   ```

4. **Set the mount point for the file system, `/datafiles` in `my-zone`.**

   ```
   zonecfg:my-zone:fs> set dir=/datafiles
   ```

5. **Specify that `/export/datafiles` in the global zone is to be mounted as `/datafiles` in `my-zone`.**

   ```
   zonecfg:my-zone:fs> set special=/export/datafiles
   ```

6. **Set the file system type.**

   ```
   zonecfg:my-zone:fs> set type=lofs
   ```

7. **End the specification.**

   ```
   zonecfg:my-zone:fs> end
   ```

8 Verify and commit the configuration.

```
zonecfg:my-zone> verify
zonecfg:my-zone> commit
```

More Information

Temporary Mounts

You can add LOFS file system mounts from the global zone without rebooting the non-global zone:

```
global# mount -F lofs /export/datafiles /export/my-zone/root/datafiles
```

To make this mount occur each time the zone boots, the zone's configuration must be modified using the `zonecfg` command.

▼ How to Delegate a ZFS Dataset to a Non-Global Zone

Use this procedure to delegate a ZFS dataset to a non-global zone.

You must be the global administrator or a user granted the appropriate authorizations in the global zone to perform this procedure.

1 Be , or have equivalent authorizations.

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 From the global zone, create a new ZFS file system named `fs2` on an existing ZFS pool named `poolA`:

```
global# zfs create poolA/fs2
```

3 Set the `mountpoint` property for the `poolA/fs2` file system to `/fs-del/fs2`.

```
global# zfs set mountpoint=/fs-del/fs2 poolA/fs2
```

4 Verify that the source of the `mountpoint` property for this file system is now `local`.

```
global# zfs get mountpoint poolA/fs2
NAME       PROPERTY    VALUE        SOURCE
poolA/fs2  mountpoint  /fs-del/fs2  local
```

5 Use the `zonecfg` command to delegate the `poolA/fs2` file system to the zone `my-zone`:

```
# zonecfg -z my-zone
zonecfg:my-zone> add dataset
zonecfg:my-zone:dataset> set name=poolA/fs2
zonecfg:my-zone:dataset> end
```

6 **Reboot the zone and display the `zoned` property for all `poolA` file systems:**

```
global# zfs get -r zoned poolA
NAME      PROPERTY  VALUE   SOURCE
poolA     zoned     off     default
poolA/fs2 zoned     on      default
```

Note that the `zoned` property for `poolA/fs2` is set to `on`. This ZFS file system was delegated to a non-global zone, mounted in the zone, and is under zone administrator control. ZFS uses the `zoned` property to indicate that a dataset has been delegated to a non-global zone at one point in time.

Adding Non-Global Zone Access to Specific File Systems in the Global Zone

▼ How to Add Access to CD or DVD Media in a Non-Global Zone

This procedure enables you to add read-only access to CD or DVD media in a non-global zone. The Volume Management file system is used in the global zone for mounting the media. A CD or DVD can then be used to install a product in the non-global zone. This procedure uses a DVD named `jes_05q4_dvd`.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Determine whether the Volume Management file system is running in the global zone.**

```
global# svcs volfs
STATE          STIME    FMRI
online         Sep_29   svc:/system/filesystem/volfs:default
```

3 **(Optional) If the Volume Management file system is not running in the global zone, start it.**

```
global# svcadm volfs enable
```

4 **Insert the media.**

5 **Check for media in the drive.**

```
global# volcheck
```

6 **Test whether the DVD is automounted.**

```
global# ls /cdrom
```

You will see a display similar to the following:

```
cdrom   cdrom1   jes_05q4_dvd
```

7 **Loopback mount the file system with the options `ro,nodevices` (read-only and no devices) in the non-global zone.**

```
global# zonecfg -z my-zone
zonecfg:my-zone> add fs
zonecfg:my-zone:fs> set dir=/cdrom
zonecfg:my-zone:fs> set special=/cdrom
zonecfg:my-zone:fs> set type=lofs
zonecfg:my-zone:fs> add options [ro,nodevices]
zonecfg:my-zone:fs> end
zonecfg:my-zone> commit
zonecfg:my-zone> exit
```

8 **Reboot the non-global zone.**

```
global# zoneadm -z my-zone reboot
```

9 **Use the `zoneadm list` command with the `-v` option to verify the status.**

```
global# zoneadm list -v
```

You will see a display that is similar to the following:

```
ID  NAME     STATUS       PATH                           BRAND     IP
 0  global   running      /                              ipkg     shared
 1  my-zone  running      /zones/my-zone                 ipkg     shared
```

10 **Log in to the non-global zone.**

```
global# zlogin my-zone
```

11 **Verify the DVD-ROM mount.**

```
my-zone# ls /cdrom
```

You will see a display similar to this:

```
cdrom   cdrom1   jes_05q4_dvd
```

12 **Install the product as described in the product installation guide.**

13 **Exit the non-global zone.**

```
my-zone# exit
```

Tip – You might want to retain the `/cdrom` file system in your non-global zone. The mount will always reflect the current contents of the CD-ROM drive, or an empty directory if the drive is empty.

---

14 **(Optional) If you want to remove the `/cdrom` file system from the non-global zone, use the following procedure.**

```
global# zonecfg -z my-zone
zonecfg:my-zone> remove fs dir=/cdrom
zonecfg:my-zone> commit
zonecfg:my-zone> exit
```

▼ How to Export Home Directories in the Global Zone Into a Non-Global Zone

This procedure is used to export home directories or other file systems from the global zone into non-global zones on the same system.

You must be the global administrator or a user granted the required rights profile in the global zone to perform this procedure.

1 **Be superuser, or have required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Add the loopback-mounted filesystem.**

```
global# zonecfg -z my-zone
zonecfg:my-zone> add fs
zonecfg:my-zone:fs> set dir=/export/home
zonecfg:my-zone:fs> set special=/export/home
zonecfg:my-zone:fs> set type=lofs
zonecfg:my-zone:fs> set options=nodevices
zonecfg:my-zone:fs> end
zonecfg:my-zone> commit
zonecfg:my-zone> exit
```

3 **Add the following line to the zone's `/etc/auto_home` file:**

```
$HOST:/export/home/&
```

Using IP Network Multipathing on an Oracle Solaris System With Zones Installed

▼ How to Use IP Network Multipathing in Exclusive-IP Non-Global Zones

IP Network Multipathing (IPMP) in an exclusive-IP zone is configured as it is in the global zone.

You can configure one or more physical interfaces into an IP multipathing group, or IPMP group. After configuring IPMP, the system automatically monitors the interfaces in the IPMP group for failure. If an interface in the group fails or is removed for maintenance, IPMP automatically migrates, or fails over, the failed interface's IP addresses. The recipient of these addresses is a functioning interface in the failed interface's IPMP group. The failover feature of IPMP preserves connectivity and prevents disruption of any existing connections. Additionally, IPMP improves overall network performance by automatically spreading out network traffic across the set of interfaces in the IPMP group. This process is called load spreading.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Configure IPMP groups as described in "Configuring IPMP Groups" in *System Administration Guide: Network Interfaces and Network Virtualization*.**

▼ How to Extend IP Network Multipathing Functionality to Shared-IP Non-Global Zones

Use this procedure to configure IPMP in the global zone and extend the IPMP functionality to non-global zones.

Each address, or logical interface, should be associated with a non-global zone when you configure the zone. See "Using the `zonecfg` Command" on page 218 and "How to Configure the Zone" on page 238 for instructions.

This procedure accomplishes the following:

- The cards `bge0` and `hme0` are configured together in a group.
- Address 192.168.0.1 is associated with the non-global zone *my-zone*.
- The `bge0` card is set as the physical interface. Thus, the IP address is hosted in the group that contains the `bge0` and `hme0` cards.

In a running zone, you can use the `ifconfig` command to make the association. See "Shared-IP Network Interfaces" on page 317 and the `ifconfig(1M)` man page.

You must be the global administrator or a user granted the appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 In the global zone, configure IPMP groups as described in "Configuring IPMP Groups" in *System Administration Guide: Network Interfaces and Network Virtualization*.

3 Use the `zonecfg` command to configure the zone. When you configure the `net` resource, add address `192.168.0.1` and physical interface `bge0` to the zone *my-zone*:

```
zonecfg:my-zone> add net
zonecfg:my-zone:net> set address=192.168.0.1
zonecfg:my-zone:net> set physical=bge0
zonecfg:my-zone:net> end
```

Only `bge0` would be visible in non-global zone *my-zone*.

More Information

If `bge0` Subsequently Fails

If `bge0` subsequently fails and the `bge0` data addresses fail over to `hme0` in the global zone, the *my-zone* addresses migrate as well.

If address `192.168.0.1` moves to `hme0`, then only `hme0` would now be visible in non-global zone *my-zone*. This card would be associated with address `192.168.0.1`, and `bge0` would no longer be visible.

Administering Data-Links in Exclusive-IP Non-Global Zones

The `dladm` command is used from the global zone to administer data-links.

▼ How to Use `dladm show-linkprop`

The `dladm` command can be used with the `show-linkprop` subcommand to show the assignment of data-links to running exclusive-IP zones.

You must be the global administrator or a user granted the appropriate authorizations in the global zone to administer data-links.

1 Be superuser, or have equivalent authorizations.

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Show the assignment of data-links on the system.

```
global# dladm show-linkprop
```

Example 25–1

Using `dladm` With the `show-linkprop` subcommand

1. In the first screen, zone `49bge`, which is assigned `bge0` has not been booted

```
global# dladm show-linkprop
LINK          PROPERTY        VALUE          DEFAULT        POSSIBLE
bge0          zone            --             --             --
ath0          channel         6              --             --
ath0          powermode       ?              off            off,fast,max
ath0          radio           ?              on             on,off
ath0          speed           11             --
1,2,5.5,6,9,11,12,18,24,36,48,54
ath0          zone            --             --             --
```

2. Zone 49bge is booted.

```
global# zoneadm -z 49bge boot
```

3. The command `dladm show-linkprop` is run again. Note that the `bge0` link is now assigned to `49bge`.

```
global# dladm show-linkprop
LINK          PROPERTY        VALUE          DEFAULT        POSSIBLE
bge0          zone            49bge          --             --
ath0          channel         6              --             --
ath0          powermode       ?              off            off,fast,max
ath0          radio           ?              on             on,off
ath0          speed           11             --
1,2,5.5,6,9,11,12,18,24,36,48,54
ath0          zone            --             --             --
```

▼ How to Use `dladm set-linkprop`

The `dladm` command can be used with the `set-linkprop` subcommand to temporarily assign data-links to running exclusive-IP zones. Persistent assignment must be made through the `zonecfg` command.

You must be the global administrator or a user granted the appropriate authorizations in the global zone to administer data-links.

1 Be superuser, or have equivalent authorizations.

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Use `dladm set-linkprop` with the `-t` to add `bge0` to a running zone called `excl`.

```
global# dladm set-linkprop -t -p zone=excl bge0
LINK          PROPERTY        VALUE          DEFAULT        POSSIBLE
bge0          zone            excl           --             --
```

Tip – The `-p` option produces a display using a stable machine-parseable format.

▼ How to Use `dladm reset-linkprop`

The `dladm` command can be used with the `reset-linkprop` subcommand to reset the `bge0` link value to unassigned.

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Use `dladm reset-linkprop` with the `-t` to undo the zone assignment of the `bge0` device.**

```
global# dladm set-linkprop -t -p zone=excl bge0
LINK         PROPERTY        VALUE          DEFAULT        POSSIBLE
bge0         zone            excl           --             --
```

Tip – The `-p` option produces a display using a stable machine-parseable format.

Troubleshooting If the running zone is using the device, the reassignment fails and an error message is displayed. See "Exclusive-IP Zone Is Using Device, so `dladm reset-linkprop` Fails" on page 357.

Using the Fair Share Scheduler on an Oracle Solaris System With Zones Installed

Limits specified through the `prctl` command are not persistent. The limits are only in effect until the system is rebooted. To set shares in a zone permanently, see "How to Configure the Zone" on page 238 and "How to Set `zone.cpu-shares` in the Global Zone" on page 248.

▼ How to Set FSS Shares in the Global Zone Using the `prctl` Command

The global zone is given one share by default. You can use this procedure to change the default allocation. Note that you must reset shares allocated through the `prctl` command whenever you reboot the system.

You must be the global administrator or a user granted the appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have equivalent authorizations.**

   For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Use the `prctl` utility to assign two shares to the global zone:**

```
# prctl -n zone.cpu-shares -v 2 -r -i zone global
```

3 **(Optional) To verify the number of shares assigned to the global zone, type:**

```
# prctl -n zone.cpu-shares -i zone global
```

See Also For more information on the `prctl` utility, see the `prctl(1)` man page.

▼ How to Change the `zone.cpu-shares` Value in a Zone Dynamically

This procedure can be used in the global zone or in a non-global zone.

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Use the `prctl` command to specify a new value for `cpu-shares`.**

```
# prctl  -n zone.cpu-shares -r -v value -i zone zonename
```

idtype is either the *zonename* or the *zoneid*. *value* is the new value.

Using Rights Profiles in Zone Administration

This section covers tasks associated with using rights profiles in non-global zones.

Backing Up an OracleSolaris System With Installed Zones

The following procedures can be used to back up files in zones. Remember to also back up the zones' configuration files.

▼ How to Use `find` and `cpio` to Perform Backups

1 **Be superuser, or have equivalent authorizations.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Change directories to the root directory.**

```
global# cd /
```

3 **Back up `my-zone` files that are not loopback mounted to `/backup/my-zone.cpio`.**

```
global# find zones/my-zone -fstype lofs -prune -o -local
 | cpio -oc -O /backup/my-zone.cpio      type as one line
```

4 **Verify the results.**

```
global# ls -l backup/my-zone.cpio
```

You will see a display similar to the following:

```
-rwxr-xr-x   1 root     root     99680256 Aug 10 16:13 backup/my-zone.cpio
```

▼ How to Print a Copy of a Zone Configuration

You should create backup files of your non-global zone configurations. You can use the backups to recreate the zones later if necessary. Create the copy of the zone's configuration after you have logged in to the zone for the first time and have responded to the `sysidtool` questions. This procedure uses a zone named `my-zone` and a backup file named `my-zone.config` to illustrate the process.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Print the configuration for the zone `my-zone` to a file named `my-zone.config`.**

```
global# zonecfg -z my-zone export > my-zone.config
```

Restoring a Non-Global Zone

▼ How to Restore an Individual Non-Global Zone

You can use the backup files of your non-global zone configurations to restore non-global zones, if necessary. This procedure uses a zone named `my-zone` and a backup file named `my-zone.config` to illustrate the process of restoring a zone.

1 **Be superuser, or have the required rights profile.**

For more information about roles, see "Configuring and Using RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Specify that `my-zone.config` be used as the `zonecfg` command file to recreate the zone `my-zone`.**

```
global# zonecfg -z my-zone -f my-zone.config
```

3 **Install the zone.**

```
global# zoneadm -z my-zone install
```

4 **To prevent the system from displaying the `sysidtool` questions upon initial zone login, delete the file *zonepath*`/root/etc/.UNCONFIGURED`, for example:**

```
global# rm /zones/my-zone/root/etc/.UNCONFIGURED
```

5 **If you have any zone-specific files to restore, such as application data, manually restore (and possibly hand-merge) files from a backup into the newly created zone's root file system.**

CHAPTER 26

Troubleshooting Miscellaneous Oracle Solaris Zones Problems

This chapter contains zones troubleshooting information.

Updating Non-Global Zones in the Oracle Solaris 11 Express Release

Currently, `ipkg` zones employ manual syncing. The zones do not automatically update when `pkg image-update` is executed on the system. You must manually update the zones after rebooting to keep them in sync with the global zone.

To manually update zones after rebooting, use the`zoneadm detach` and `attach` commands with the `-u` option.

Run `pkg image-update` and reboot onto the new image, then detach the zone and use `zoneadm attach` with the `-u` option. See "About Migrating a Zone" on page 294 for more information on these commands.

Also note that the dry-run installation feature of `pkg` install does not work in Oracle Solaris Zones.

Exclusive-IP Zone Is Using Device, so `dladm reset-linkprop` Fails

If the following error message is displayed:

```
dladm: warning: cannot reset link property 'zone' on 'bge0': operation failed
```

Referring to "How to Use `dladm reset-linkprop`" on page 353, the attempt to use `dladm reset-linkprop` failed. The running zone `excl` is using the device, which was assigned by executing `ifconfig bge0 plumb` inside the zone.

To reset the value, use the procedure `ifconfig bge0 unplumb` inside the zone and rerun the `dladm` command.

Incorrect Privilege Set Specified in Zone Configuration

If the zone's privilege set contains a disallowed privilege, is missing a required privilege, or includes an unknown privilege name, an attempt to verify, ready, or boot the zone will fail with an error message such as the following:

```
zonecfg:zone5> set limitpriv="basic"
.
.
.
global# zoneadm -z zone5 boot
     required privilege "sys_mount" is missing from the zone's privilege set
     zoneadm: zone zone5 failed to verify
```

Zone Administrator Mounting Over File Systems Populated by the Global Zone

The presence of files within a file system hierarchy when a non-global zone is first booted indicates that the file system data is managed by the global zone. When the non-global zone was installed, a number of the packaging files in the global zone were duplicated inside the zone. These files must reside under the `zonepath` directly. If the files reside under a file system created by a zone administrator on disk devices or ZFS datasets added to the zone, packaging problems could occur.

The issue with storing any of the file system data that is managed by the global zone in a zone-local file system can be described by using ZFS as an example. If a ZFS dataset has been delegated to a non-global zone, the zone administrator should not use that dataset to store any of the file system data that is managed by the global zone. The configuration could not be upgraded correctly.

For example, a ZFS delegated dataset should not be used as a `/var` file system. The Oracle Solaris operating system delivers core packages that install components into `/var`. These packages have to access `/var` when they are upgraded, which is not possible if `/var` is mounted on a delegated ZFS dataset.

File system mounts under parts of the hierarchy controlled by the global zone are supported. For example, if an empty `/usr/local` directory exists in the global zone, the zone administrator can mount other contents under that directory.

You can use a delegated ZFS dataset for file systems that do not need to be accessed during upgrade, such as `/export` in the non-global zone.

`netmasks` Warning Displayed When Booting Zone

If you see the following message when you boot the zone as described in "How to Boot a Zone" on page 269:

```
# zoneadm -z my-zone boot
zoneadm: zone 'my-zone': WARNING: hme0:1: no matching subnet
    found in netmasks(4) for 192.168.0.1; using default of
    255.255.255.0.
```

The message is only a warning, and the command has succeeded. The message indicates that the system was unable to find the `netmask` to be used for the IP address specified in the zone's configuration.

To stop the warning from displaying on subsequent reboots, ensure that the correct `netmasks` databases are listed in the `/etc/nsswitch.conf` file in the global zone and that at least one of these databases contains the subnet and `netmasks` to be used for the zone `my-zone`.

For example, if the `/etc/inet/netmasks` file and the local NIS database are used for resolving `netmasks` in the global zone, the appropriate entry in `/etc/nsswitch.conf` is as follows:

```
netmasks: files nis
```

The subnet and corresponding netmask information for the zone `my-zone` can then be added to `/etc/inet/netmasks` for subsequent use.

For more information about the `netmasks` command, see the `netmasks(4)` man page.

Zone Does Not Halt

In the event that the system state associated with the zone cannot be destroyed, the halt operation will fail halfway. This leaves the zone in an intermediate state, somewhere between running and installed. In this state there are no active user processes or kernel threads, and none can be created. When the halt operation fails, you must manually intervene to complete the process.

The most common cause of a failure is the inability of the system to unmount all file systems. Unlike a traditional Oracle Solaris system shutdown, which destroys the system state, zones must ensure that no mounts performed while booting the zone or during zone operation remain once the zone has been halted. Even though `zoneadm` makes sure that there are no processes executing in the zone, the unmount operation can fail if processes in the global zone have open files in the zone. Use the tools described in the `proc`(1) (see `pfiles`) and `fuser`(1M) man pages to find these processes and take appropriate action. After these processes have been dealt with, reinvoking `zoneadm halt` should completely halt the zone.

For a zone that cannot be halted, you can migrate a zone that has not been detached by using the `zoneadm attach -F` option to force the attach without a validation. The target system must be properly configured to host the zone. An incorrect configuration could result in undefined behavior. Moreover, there is no way to know the state of the files within the zone.

PART III

Oracle Solaris 10 Zones

Oracle Solaris 10 Zones are `solaris10`branded zones that host x86 and SPARC Solaris 10 10/09 (or later released Oracle Solaris 10 update) user environments running on the Oracle Solaris 11 Express kernel. Note that it is possible to use an earlier Solaris 10 release if you first install the kernel patch 141444-09 (SPARC) or 141445-09 (x86/x64), or later version, on the original system.

CHAPTER 27

Introduction to Oracle Solaris 10 Zones

BrandZ provides the framework to create branded zones, which are used to run applications that cannot be run in an Oracle Solaris 11 Express environment. The brand described here is the `solaris10` brand, Oracle Solaris 10 Zones. Workloads running within these `solaris10` branded zones can take advantage of the enhancements made to the Oracle Solaris kernel and utilize some of the innovative technologies available only on the Oracle Solaris 11 Express release, such as virtual NICs (VNICs) and ZFS deduplication.

Note – If you want to create `solaris10` branded zones now, go to Chapter 28, "Assessing an Oracle Solaris 10 System and Creating an Archive."

About the `solaris10` Brand

The `solaris10` branded zone, described in the `solaris10(5)` man page, is a complete runtime environment for Oracle Solaris 10 applications on SPARC and x86 machines running the Oracle Solaris 10 10/09 operating system or later released update. If you are running an Oracle Solaris 10 release earlier than Oracle Solaris 10 10/09, it is possible to use the earlier update release if you first install the kernel patch 141444-09 (SPARC) or 141445-09 (x86/x64), or later version, on the original system. You must install the patch before you create the archive that will be used to install the zone. It is the kernel patch of the release that is the prerequisite for migration to an Oracle Solaris 10 Container, not the full Oracle Solaris 10 10/09 release. For information regarding patches, contact your support provider.

The brand is supported on all `sun4v`, `sun4u`, and x86 architecture machines that the Oracle Solaris 11 Express release has defined as supported platforms. The brand supports the execution of 32-bit and 64-bit Oracle Solaris 10 applications.

The brand includes the tools required to install an Oracle Solaris 10 system image into a non-global zone. You cannot install a `solaris10` branded zone directly from Oracle Solaris 10 media. A physical-to-virtual (P2V) capability is used to directly migrate an existing system into a non-global zone on a target system. The brand also supports the tools used to migrate an Oracle Solaris 10 `native` zone into a `solaris10` brand non-global zone. The virtual-to-virtual (V2V) process for migrating an Oracle Solaris 10 `native` non-global zone into a `solaris10` branded zone supports the same archive formats as P2V. See Chapter 29, "(Optional) Migrating an Oracle Solaris 10 `native` Non-Global Zone Into an Oracle Solaris 10 Container," for more information.

The `solaris10` brand supports the whole root non-global zone model. All of the required Oracle Solaris 10 software and any additional packages are installed into the private file systems of the zone.

The non-global zone must reside on its own ZFS dataset; only ZFS is supported. The ZFS dataset will be created automatically when the zone is installed or attached. If a ZFS dataset cannot be created, the zone will not install or attach. Note that the parent directory of the zone path must also be a ZFS dataset, or the file system creation will fail.

Any application or program that executes in a `native` Oracle Solaris 10 non-global zone should also work in a `solaris10` branded zone.

Note – You can create and install `solaris10` branded zones on an Oracle Solaris Trusted Extensions system that has labels enabled, but you can only boot branded zones on this system configuration *if* the brand being booted is the `labeled` brand. Customers using Oracle Solaris Trusted Extensions on Oracle Solaris 10 systems must transition to a certified Oracle Solaris system configuration.

SVR4 Packaging and Patching in Oracle Solaris 10 Zones

About Using Packaging and Patching in `solaris10` Branded Zones

The SVR4 package metadata is available inside the zone, and the package and patch commands work correctly. For proper operation, note that you *must* install patches 119254-75 (SPARC) or 119255-75 (x86/x64), or later versions, on your Oracle Solaris 10 system *before* the archive is created. Contact your support provider for information regarding patches.

Because the zones are whole root zones, all packaging and patch operations are successful, although the kernel components of the package or patch are not used. For information on SVR4 packaging used in `solaris10` `native` zones, see "Chapter 25, About Packages on an Solaris System With Zones Installed (Overview)" and "Chapter 26, Adding and Removing Packages and Patches on a Solaris System With Zones Installed (Tasks)" in *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones*. This is the Oracle Solaris 10 version of the guide.

About Performing Package and Patch Operations Remotely

For patch operations initiated from within Oracle Solaris 10 Zones, if the remote system is another Oracle Solaris 10 Container, the patching operation works correctly. However, if the remote system is a miniroot or an Oracle Solaris 10 system that is not an Oracle Solaris 10 Container, the operation will produce undefined results. Similarly, the patch tools will produce undefined results if used to patch Oracle Solaris 10 Zones from miniroots or systems instead of Oracle Solaris 10 Zones.

The `patchadd` and `patchrm` tools allow administrators to specify alternate roots when running patch operations. This capability allows administrators to patch remote systems, such as Oracle Solaris 10 miniroots and Oracle Solaris 10 physical systems, which have `root` directories visible over NFS.

For example, if the root directory of an Oracle Solaris 10 system is NFS-mounted onto a local system's `/net/a-system` directory, then the remote Oracle Solaris 10 system could be patched from the local system.

To install patch 142900-04 (or later version) on the remote system:

```
# patchadd -R /net/a-system 142900-04
```

For more information, see the following man pages in the *man pages section 1M: System Administration Commands*:

- `patchadd`(1M), the `-R` and `-C` options
- `patchrm`(1M)

General Zones Concepts

You should be familiar with the following resource management and zones concepts, which are discussed in Part I, "Oracle Solaris Resource Management," and Part II, "Oracle Solaris Zones," of this guide.

- Supported and unsupported features
- Resource controls that enable the administrator to control how applications use available system resources
- Commands used to configure, install, and administer zones, primarily `zonecfg`, `zoneadm`, and `zlogin`
- `zonecfg` resources and property types
- The global zone and the non-global zone
- The whole-root non-global zone model

- Authorizations granted through the `zonecfg` utility
- The global administrator and the zone administrator
- The zone state model
- The zone isolation characteristics
- Privileges
- Networking
- Zone shared-IP and exclusive-IP types
- The use of resource management features, such as resource pools, with zones
- The fair share scheduler (FSS), a scheduling class that enables you to allocate CPU time based on shares
- The resource capping daemon (`rcapd`), which can be used from the global zone to control resident set size (RSS) usage of branded zones

About Oracle Solaris 10 Zones in This Release

Operating Limitations

A `/dev/sound` device cannot be configured into the `solaris10` branded zone.

Debugging Tools and System Call Traps

Debugging tools can be used to debug single processes inside an Oracle Solaris 10 Container.

Administrators must use the `truss` command, `mdb` command, and other debugging tools that can observe system call traps, such as the `dbx` debugging tool, from the Oracle Solaris 11 global zone if the commands will follow child processes.

Debugging commands and tools that can observe system call traps do not properly follow child processes forked from controlled processes when the commands are executed within `solaris10` branded zones. For example:

`truss -f -p` *PID*

does not follow the children of the process identified by *PID*. Attempting to follow child processes with these tools within Oracle Solaris 10 Zones can result in undefined behavior.

Additionally, the Oracle Solaris 10 `truss`, `mdb`, and other debugging commands will not observe the following `syscall` traps and their 64-bit equivalents because the traps were eliminated, re-implemented, or renumbered in Oracle Solaris 11 Express:

- access
- chmod
- chown
- creat
- dup
- fchmod
- fchown
- forkall
- fork1
- fsat
- fstat
- fxstat
- lchown
- link
- lstat
- lwp_mutex_lock
- lwp_sema_wait
- lxstat
- mkdir
- mknod
- poll
- readlink
- rename
- rmdir
- stat
- symlink
- umount
- unlink
- utime
- utimes
- xmknod
- xstat

For more information, see mdb(1) and truss(1).

Networking in Oracle Solaris 10 Zones

The following sections identify Oracle Solaris 10 networking features that are either not available in Oracle Solaris 10 Zones or are different in Oracle Solaris 10 Zones.

Networking Features That Are not Supported

- Mobile IP is not supported. This feature is not available in Oracle Solaris 11 Express.
- Automatic tunnels using the atun STREAMS module are not supported.
- The following ndd tunable parameters are not supported in a solaris10 branded zone:
  - ip_squeue_fanout
  - ip_soft_rings_cnt
  - ip_ire_pathmtu_interval
  - tcp_mdt_max_pbufs

Networking Features That Are Different

In a solaris10 branded zone with an exclusive-IP configuration, the following features are different from a physical Oracle Solaris 10 system:

- In a solaris10 branded zone, an autopush configuration will be ignored when the tcp, udp, or icmp sockets are open. These sockets are mapped to modules instead of STREAMS devices by default. To use autopush, explicitly map these sockets to STREAMS-based devices by using the soconfig and sock2path.d utilities described in the soconfig(1M) and sock2path.d(4) man pages.
- In a solaris10 branded zone, you must install the patch to support /dev/net links in the Data Link Provider Interface (DLPI) library, which is described in thelibdlpi(3LIB) man page.
  - 145923-01 or later version (SPARC)
  - 145924-01 or later version (x86/x64)

  Contact your support provider for information regarding patches.

  Applications that do not use either the patched libdlpi or libpcap versions 1.0.0 or higher libraries will not be able to access /dev/net links.
- Because IP Network Multipathing (IPMP) in Oracle Solaris 10 Zones is based on the Oracle Solaris 11 Express release, there are differences in the output of the ifconfig command when compared to the command output in the Oracle Solaris 10 operating system. However, the documented features of the ifconfig command and IPMP have not changed. Therefore, Oracle Solaris 10 applications that use the documented interfaces will continue to work in Oracle Solaris 10 Zones without modification.

  The following example shows ifconfig command output in a solaris10 branded zone for an IPMP group ipmp0 with data address 198.162.1.3 and the underlying interfaces e1000g1 and e1000g2, with test addresses 198.162.1.1 and 198.162.1.2, respectively.

  ```
  % ifconfig -a
  e1000g1:
  flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
  mtu 1500 index 8
          inet 198.162.1.1 netmask ffffff00 broadcast 198.162.1.255
  ```

```
        ether 0:11:22:45:40:a0
e1000g2:
flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 9
        inet 198.162.1.2 netmask ffffff00 broadcast 198.162.1.255
        ether 0:11:22:45:40:a1
ipmp0: flags=8011000803<UP,BROADCAST,MULTICAST,IPv4,FAILED,IPMP> mtu 68
index 10
        inet 198.162.1.3 netmask ffffff00 broadcast 198.162.1.255
        groupname ipmp0
```

- Unlike the display produced on an Oracle Solaris 10 system, the `ifconfig` command in an Oracle Solaris 10 Container does not show the binding of the underlying interfaces to IP addresses. This information can be obtained by using the `arp` command with the `-an` options.
- If an interface is plumbed for IPv6 and address autoconfiguration succeeds, then the interface is given its own global address. In Oracle Solaris 10, each physical interface in an IPMP group will have its own global address, and the IPMP group will have as many global addresses as there are interfaces. In an Oracle Solaris 10 Container, only the IPMP interface will have its own global address. The underlying interfaces will not have their own global addresses.
- Unlike the Oracle Solaris 10 operating system, if there is only one interface in an IPMP group, then its test address and its data address cannot be the same.

See the `arp`(1M) and `ifconfig`(1M) man pages, and "IP Network Multipathing in Exclusive-IP Zones" on page 319.

If `native` Non-Global Zones Are Installed

An additional step in the P2V process occurs when there are `native` zones on the Oracle Solaris 10 10/09 (or later released update) source physical system. Because zones do not nest, the P2V process on these systems makes the existing zones unusable inside the branded zone. The existing zones are detected when the zone is installed, and a warning is issued indicating that any nested zones will not be usable and that the disk space could be recovered. Those zones can be migrated first using the V2V feature described in Chapter 29, "(Optional) Migrating an Oracle Solaris 10 `native` Non-Global Zone Into an Oracle Solaris 10 Container."

If you apply the kernel patch on a system running an earlier release, apply the patch before you migrate the existing zones.

CHAPTER 28

Assessing an Oracle Solaris 10 System and Creating an Archive

This chapter discusses obtaining information about the Oracle Solaris 10 10/09 (or later released update) system and creating the archive of the system. A physical-to-virtual (P2V) capability is used to directly migrate an existing Oracle Solaris system into a non-global zone on a target system. Information on required packages on the target system is also provided.

Source and Target System Prerequisites

Enabling Oracle Solaris 10 Package and Patch Tools

To use the Oracle Solaris 10 package and patch tools in your Oracle Solaris 10 Container, install patches 119254-75 (SPARC) and 119255-75 (x86/x64) on your source system *before* the image is created. The P2V process will work without the patches, but the package and patch tools will not work properly within the `solaris10` branded zone.

Installing the Required Oracle Solaris Package on the Target System

To use Oracle Solaris 10 Zones on your system, the `system/zones/brand/s10` package must be installed on the system running Oracle Solaris 11 Express.

For more information on the repository, see "Image Packaging System Software on Systems Running the Oracle Solaris 11 Express Release" on page 301.

For package installation instructions, see *Oracle Solaris 11 Express Image Packaging System Guide*.

Assess the System To Be Migrated

An existing Oracle Solaris 10 10/09 system (or later released Solaris 10 update) can be directly migrated into a `solaris10` branded zone on an Oracle Solaris 11 Express system.

Depending on the services performed by the original system, the global administrator or a user granted the appropriate authorizations might need to manually customize the zone after it has been installed. For example, the privileges assigned to the zone might need to be modified. This is not done automatically. Also, because not all system services work inside zones, not every Oracle Solaris 10 system is a good candidate for migration into a zone.

To begin, examine the source system and collect needed information.

- Obtain the hostname:

  ```
  # hostname
  ```
- Obtain the host ID:

  ```
  # hostid
  ```

  Also see "Host ID in Zones" on page 213.
- Obtain the RPC `domainname`:

  ```
  # domainname
  ```
- Obtain the root password.
- View the software being run on the system:

  ```
  # ps -eaf
  ```
- Check the networking utilized on the system:

  ```
  # ifconfig -a
  ```
- View the storage utilized, for example, by viewing the contents of `/etc/vfstab`.
- View the amount of local disk storage in use, which determines the size of the archive:

  ```
  # df -k
  ```
- Determine the packages and patches that are on the system. To use the Oracle Solaris 10 package and patch tools in your Oracle Solaris 10 Container, ensure that patch 119254-75 (SPARC) or 119255-75 (x86/x64) has been installed on your Solaris 10 system before the image is created.
- Examine the contents of `/etc/system` to view the update level of the Oracle Solaris 10 system.

Note – If there are any `native` non-global zones on the system to be migrated, these zones must either be deleted, or be archived and moved into zones on the new target system first. For a sparse root zone, the archive must be made with the zone in the ready state. For additional information on migration, see Chapter 29, "(Optional) Migrating an Oracle Solaris 10 `native` Non-Global Zone Into an Oracle Solaris 10 Container." For additional information on sparse root zones, see "Zones Overview" on page 190 in the Oracle Solaris 10 documentation.

Creating the Image for Directly Migrating Oracle Solaris 10 Systems Into Zones

You can use the Oracle Solaris Flash archiving tools to create an image of an installed system that can be migrated into a zone.

The system can be fully configured with all of the software that will be run in the zone before the image is created. This image is then used by the installer when the zone is installed.

Caution – If you create an Oracle Solaris Flash archive, or `flar`, of an Oracle Solaris 10 system with a ZFS root, then by default, the `flar` will actually be a ZFS send stream, which can be used to recreate the root pool. This image *cannot* be used to install an Oracle Solaris 10 Container. You must create the `flar` with an explicit `cpio` or `pax` archive when the system has a ZFS root.

To create the `flar`, use the `flarcreate` command with the `-L archiver` option, specifying `cpio` or `pax` as the method to archive the files. See Step 4 in the next procedure. Also see "Other Archive Creation Methods" on page 374.

▼ How to Use `flarcreate` to Create the Image

Use the `flarcreate` command described in the `flarcreate`(1M) Oracle Solaris 10 man page to create the system image. This example procedure uses NFS to place the flash archive on the target Oracle Solaris 11 Express system, but you could use any method to move the files.

You must be the global administrator or a user with the appropriate authorizations in the global zone to perform this procedure.

1 **Be superuser, or have the required rights profile.**

2 **Log into the source Oracle Solaris 10 system to be archived.**

3 **Change directories to the root directory.**

```
# cd /
```

4 **Use `flarcreate` to create a flash archive image file named `s10-system` on the source system, and place the archive onto the target Oracle Solaris 11 Express system:**

```
source-system # flarcreate -S -n s10-system -L cpio /net/target/export/s10-system.flar
Determining which filesystems will be included in the archive...
Creating the archive...
cpio: File size of "etc/mnttab" has
increased by 435
2068650 blocks
1 error(s)
Archive creation complete.
```

Tip – In some cases, `flarcreate` can display errors from the `cpio` command. Most commonly, these are messages such as `File size of etc/mnttab has increased by 33`. When these messages pertain to log files or files that reflect system state, they can be ignored. Be sure to review all error messages thoroughly.

Other Archive Creation Methods

You can use alternate methods for creating the archive. The installer can accept the following archive formats:

- `cpio` archives
- `gzip` compressed `cpio` archives
- `bzip2` compressed `cpio` archives
- `pax` archives created with the `-x xustar` (`XUSTAR`) format
- `ufsdump` level zero (full) backups

Additionally, the installer can only accept a directory of files created by using an archiving utility that saves and restores file permissions, ownership, and links.

For more information, see the `cpio`(1), `pax`(1), `bzip2`(1), `gzip`(1), and `ufsdump`(1M) man pages.

Note – If you use a method other than flash archive for creating an archive for P2V, you must unmount the processor-dependent `libc.so.1` `lofs`-mounted hardware capabilities (`hwcap`) library on the source system before you create the archive. Otherwise, the zone installed with the archive might not boot on the target system. After you have created the archive, you can remount the proper hardware capabilities library on top of `/lib/libc.so.1` by using `lofs` and the mount `-O` option.

```
source-system# unmount /lib/libc.so.1
source-system# mount -O -F lofs /lib/libc.so.1
```

Host ID Emulation

When applications are migrated from a standalone Oracle Solaris system into a zone on a new system, the `hostid` changes to be the `hostid` of the new machine.

In some cases, applications depend on the original `hostid`, and it is not possible to update the application configuration. In these cases, the zone can be configured to use the `hostid` of the original system. This is done by setting a `zonecfg` property to specify the `hostid`, as described in "How to Configure the Zone" on page 238. The value used should be the output of the `hostid` command as run on the original system. To view the `hostid` in an installed zone, also use the `hostid` command.

For more information about host IDs, see `hostid(1)`.

CHAPTER 29

(Optional) Migrating an Oracle Solaris 10 native Non-Global Zone Into an Oracle Solaris 10 Container

This chapter describes migrating a native zone on an Oracle Solaris 10 10/09 (or later released update) system into an Oracle Solaris 10 Container on a system running the Ooracle Solaris 11Express release.

Only read this chapter if there are any native non-global zones on the system that you want to migrate. These zones must be archived and moved into branded zones on the new target system first.

Archive Considerations

A sparse root zone on an Oracle Solaris 10 system must be converted to a whole root model for the solaris10 branded zone migration. A sparse root zone must be in the ready state on the source system before the V2V process. This will mount any inherited-pkg-dir resources before the archive is created. See "Zones Overview" on page 190 in the Oracle Solaris 10 version of this guide for more information on these concepts.

The process will be very similar to the existing zone migration feature. The zone's brand will be changed as part of the process.

Overview of the solaris10 Zone Migration Process

The virtual-to-virtual (V2V) process for migrating an Oracle Solaris 10 native zone to a solaris10 branded zone supports the same archive formats as P2V. This process uses the zoneadm attach subcommand, which is the existing interface for migrating zones from one system to another. The solaris10 brand attach subcommand uses the following options, which correspond to the same options in the install subcommand.

Option	Description
-a *path*	Specifies a path to an archive to unpack into the zone. Full flash archive and pax, cpio, gzip compressed cpio, bzip compressed cpio, and level 0 ufsdump are supported.
-d *path*	Specifies a path to a tree of files as the source for the installation.
-d –	Use the -d option with the dash parameter to direct that the existing directory layout be used in the zonepath. Thus, if the administrator manually sets up the zonepath directory before the installation, the -d – option can be used to indicate that the directory already exists.

About Detaching and Attaching the solaris10 Branded Zone

A solaris10 brand zone can be migrated to an Oracle Solaris host by configuring the zone on the target system, then using the zoneadm command with the detach and attach subcommands and either the -a option to attach an archive or the -d option to specify a zonepath. This process is described in "About Migrating a Zone" on page 294 and "How to Migrate A Non-Global Zone" on page 295.

Migrating a solaris10 Branded Zone

The zonecfg and zoneadm commands can be used to migrate an existing non-global zone from one system to another. The zone is halted and detached from its current host. The zonepath is moved to the target host, where it is attached.

The zoneadm detach process creates the information necessary to attach the zone on a different system. The zoneadm attach process verifies that the target machine has the correct configuration to host the zone.

Because there are several ways to make the zonepath available on the new host, the actual movement of the zonepath from one system to another is a manual process that is performed by the global administrator.

When attached to the new system, the zone is in the installed state.

EXAMPLE 29–1 Sample attach Command

```
host2# zoneadm -z zonename attach -a /net/machine_name/s10-system.flar
```

Migrating an Existing Zone on an Oracle Solaris 10 System

Before a physical system can be migrated, any existing non-global zones on the system must be archived and moved into zones on the new target system first.

How to Migrate an Existing `native` Non-Global Zone

Use the V2V process to migrate an existing zone on your Solaris 10 system to a `solaris10` brand zone on aa system running the Oracle Solaris 11 Express release.

1 **Print the existing zone's configuration. You will need this information to recreate the zone on the destination system:**

```
source# zonecfg -z my-zone info
zonename: my-zone
zonepath: /zones/my-zone
brand: native
autoboot: false
bootargs:
pool:
limitpriv:
scheduling-class:
ip-type: shared
hostid: 1337833f
inherit-pkg-dir:
         dir: /lib
inherit-pkg-dir:
         dir: /platform
inherit-pkg-dir:
         dir: /sbin
inherit-pkg-dir:
         dir: /usr
net:
         address: 192.168.0.90
         physical: bge0
```

2 **Halt the zone:**

```
source# zoneadm -z myzone halt
```

You should not archive a running zone since the application or system data within the zone might be captured in an inconsistent state.

3 **(Optional) If the zone is a sparse root zone that has `inherit-pkg-dir` settings, then first ready the zone so that the inherited directories will be archived:**

```
source# zoneadm -s myzone ready
```

4 **Archive the zone with the `zonepath` `/zones/my-zone`.**

- **Create a `gzip` compressed `cpio` archive named `my-zone.cpio.gz` for the zone, which will still be named `my-zone` on the target system:**

```
source# cd /zones
source# find my-zone -print | cpio -oP@/ | gzip >/zones/my-zone.cpio.gz
```

- **Create the archive from within the `zonepath` if you intend to rename the zone on the target system:**

```
source# cd /zones/my-zone
source# find root -print | cpio -oP@/ | gzip >/zones/my-zone.cpio.gz
```

5 **Transfer the archive to the target Oracle Solaris 11 Express system, using any file transfer mechanism to copy the file, including:**

- The `sftp` command described in the `sftp(1)` man page
- NFS mounts
- Any other file transfer mechanism to copy the file.

6 **On the target system, recreate the zone.**

```
target# zonecfg -z my-zone
my-zone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:my-zone> create -t SUNWsolaris10
zonecfg:my-zone> set zonepath=/zones/myzone
...
```

Note – The zone's brand must be `solaris10` and the zone cannot use any `inherit-pkg-dir` settings, even if the original zone was configured as a sparse root zone. See Part II, "Oracle Solaris Zones," for information on `inherit-pkg-dir` resources.

If the destination system has different hardware, different network interfaces, or other devices or file systems that must be configured on the zone, you must update the zone's configuration. See Chapter 16, "Non-Global Zone Configuration (Overview)" Chapter 17, "Planning and Configuring Non-Global Zones (Tasks)," and "About Migrating a Zone" on page 294.

---

7 **Display the zone's configuration:**

```
target# zonecfg -z my-zone info
zonename: my-zone
zonepath: /zones/my-zone
brand: solaris10
autoboot: false
bootargs:
pool:
limitpriv:
scheduling-class:
ip-type: shared
hostid: 1337833f
```

```
net:
         address: 192.168.0.90
         physical: bge0
```

8 **Attach the zone from the archive that was created on the source system, with the archive transferred into the `/zones` directory on the destination system:**

```
target# zoneadm -z my-zone attach -a /zones/my-zone.cpio.gz
```

Once the zone installation has completed successfully, the zone is ready to boot.

You can save the zone's archive in case you need it later, or remove it from the system.

To remove the archive from the destination system:

```
target# rm /zones/myzone.cpio.gz
```

CHAPTER 30

Configuring the solaris10 Branded Zone

This chapter discusses configuring the `Solaris10` branded zone.

Preconfiguration Tasks

You will need the following:

- A supported SPARC or x86 system running the Oracle Solaris 11 Express release.
- For a zone that requires network connectivity, you will need to provide one or more unique IPv4 addresses for each shared-IP zone you want to create. You must also specify the physical interface.
- A machine running the Oracle Solaris 10 10/09 (or later released update) operating system that you want to migrate into a `solaris10` container. You can generate your own images from existing systems. The process is described in "Creating the Image for Directly Migrating Oracle Solaris 10 Systems Into Zones" on page 373.

Resources Included in the Configuration by Default

Devices, file systems, and privileges in a branded zone are included in the configuration by default.

Configured Devices in solaris10 Branded Zones

The devices supported by each zone are documented in the man pages and other documentation for that brand. The `solaris10` zone does not allow the addition of any unsupported or unrecognized devices. The framework detects any attempt to add an unsupported device. An error message is issued that indicates the zone configuration cannot be verified.

To learn more about device considerations in non-global zones, see "Device Use in Non-Global Zones" on page 320.

Privileges Defined in `solaris10` Branded Zones

Processes are restricted to a subset of privileges. Privilege restriction prevents a zone from performing operations that might affect other zones. The set of privileges limits the capabilities of privileged users within the zone.

Default, required default, optional, and prohibited privileges are defined by each brand. You can also add or remove certain privileges by using the `limitpriv` property as shown in Step 8 of "How to Configure the Zone" on page 238. The table Table 24–1 lists all of the Solaris privileges and the status of each privilege with respect to zones.

For more information about privileges, see the `ppriv(1)` man page and *System Administration Guide: Security Services*.

`solaris10` Branded Zone Configuration Process

The `zonecfg` command is used to do the following:

- Set the brand for the zone.
- Create the configuration for the `solaris10` zone.
- Verify the configuration to determine whether the specified resources and properties are allowed and internally consistent on a hypothetical system.
- Perform a brand-specific verification.

The check performed by the `zonecfg verify` command for a given configuration verifies the following:

- Ensures that a zone path is specified
- Ensures that all of the required properties for each resource are specified
- Ensures that brand requirements are met

For more information about the `zonecfg` command, see the `zonecfg(1M)` man page.

Configuring the Target Zone

Create the new zone configuration on the target system by using the `zonecfg` command.

The `zonecfg` prompt is of the following form:

```
zonecfg:zonename>
```

When you are configuring a specific resource type, such as a file system, that resource type is also included in the prompt:

```
zonecfg:zonename:fs>
```

Tip – If you know you will be using CDs or DVDs to install applications in a `solaris10` branded zone, use `add fs` to add read-only access to CD or DVD media in the global zone when you initially configure the branded zone. A CD or DVD can then be used to install a product in the branded zone. See "How to Add Access to CD or DVD Media in a Non-Global Zone" on page 347 for more information.

This procedure describes configuring a shared-IP zone.To configure an exclusive-IP zone, see "Resource Type Properties" on page 226.

▼ How to Configure a `solaris10` Branded Zone

You must be the global administrator or a user with the appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have the required rights profile.**

2. **Set up a shared-IP zone configuration with the zone name `s10-zone`.**

```
global# zonecfg -z s10-zone
```

If this is the first time you have configured this zone, you will see the following system message:

```
s10-zone: No such zone configured
Use 'create' to begin configuring a new zone.
```

3. **Create the new `solaris10` zone configuration by using the `SUNWsolaris10` template.**

```
zonecfg:s10-zone> create -t SUNWsolaris10
```

Optionally, you can also use `create` and then set the brand:

```
create
set brand=solaris10
```

4 Set the zone path, `/zones/s10-zone` in this procedure.

```
zonecfg:s10-zone> set zonepath=/zones/s10-zone
```

5 Set the autoboot value.

If set to `true`, the zone is automatically booted when the global zone is booted. Note that for the zones to autoboot, the zones service `svc:/system/zones:default` must also be enabled. The default value is `false`.

```
zonecfg:s10-zone> set autoboot=true
```

6 Add a network virtual interface.

```
zonecfg:s10-zone> add net
```

a. **Set the IP address. In this procedure, *10.6.10.233* is used.**

```
zonecfg:s10-zone:net> set address=10.6.10.233
```

b. **Set the physical device type for the network interface, the `bge` device in this procedure.**

```
zonecfg:s10-zone:net> set physical=bge0
```

c. **End the specification.**

```
zonecfg:s10-zone:net> end
```

This step can be performed more than once to add more than one network interface.

7 Add a ZFS file system shared with the global zone.

```
zonecfg:s10-zone> add fs
```

a. **Set the type to `zfs`.**

```
zonecfg:s10-zone:fs> set type=zfs
```

b. **Set the directory to mount from the global zone.**

```
zonecfg:s10-zone:fs> set special=share/zone/s10-zone
```

c. **Specify the mount point.**

```
zonecfg:s10-zone:fs> set dir=/export/shared
```

d. **End the specification.**

```
zonecfg:s10-zone:fs> end
```

This step can be performed more than once to add more than one file system.

8 Delegate a ZFS dataset named *sales* in the storage pool *tank*

```
zonecfg:my-zone> add dataset
```

a. **Specify the path to the ZFS dataset *sales*.**

```
zonecfg:my-zone> set name=tank/sales
```

b. **End the dataset specification.**

```
zonecfg:my-zone> end
```

9 **Set the `hostid` to be the `hostid` of the source system.**

```
zonecfg:my-zone> set hostid=80f0c086
```

10 **Verify the zone configuration for the zone.**

```
zonecfg:s10-zone> verify
```

11 **Commit the zone configuration for the zone.**

```
zonecfg:s10-zone> commit
```

12 **Exit the `zonecfg` command.**

```
zonecfg:s10-zone> exit
```

Note that even if you did not explicitly type `commit` at the prompt, a `commit` is automatically attempted when you type `exit` or an EOF occurs.

13 **Use the `info` subcommand to verify that the brand is set to `solaris10`.**

```
global# zonecfg -z s10-zone info
```

14 **(Optional) Use the `info` subcommand to check the `hostid`:**

```
global# zonecfg -z s10-zone info hostid
```

Next Steps

Tip – After you have configured the zone, it is a good idea to make a copy of the zone's configuration. You can use this backup to restore the zone in the future. As superuser or administrator with the correct profile, print the configuration for the zone *s10-zone* to a file. This example uses a file named `s10-zone.config`.

```
global# zonecfg -z s10-zone export > s10-zone.config
```

See Also For additional components that can be configured using `zonecfg`, see Chapter 16, "Non-Global Zone Configuration (Overview)." This guide also provides information on using the `zonecfg` command in either command-line or command-file mode. For more information about adding ZFS file systems, see "Adding ZFS File Systems to a Non-Global Zone" in *Oracle Solaris ZFS Administration Guide*.

CHAPTER 31

Installing the `solaris10` Branded Zone

This chapter covers installing a `solaris10` branded zone.

Zone Installation Images

Types of System Images

- You can use an image of an Oracle Solaris system that has been fully configured with all of the software that will be run in the zone. See “Creating the Image for Directly Migrating Oracle Solaris 10 Systems Into Zones” on page 373.

  Note that the `zoneadm install -a` command takes an archive of a physical system, *not* an archive of a zone.
- You can use an image of an existing Oracle Solaris 10 `native` zone instead of an image from a physical system. See Chapter 29, “(Optional) Migrating an Oracle Solaris 10 `native` Non-Global Zone Into an Oracle Solaris 10 Container.”

  Note that the `zoneadm attach -a` command takes an archive of a zone, *not* an archive of a physical system.

Image `sysidcfg` Status

If you created aSolaris 10 system archive from an existing system and use the `-p` (preserve `sysidcfg`) option when you install the zone, then the zone will have the same identity as the system used to create the image.

If you use the `-u` (`sys-unconfig`) option when you install the target zone, the zone produced will not have a hostname or name service configured.

Install the `solaris10` Branded Zone

The `zoneadm` command described in Part II, "Oracle Solaris Zones," and in the `zoneadm(1M)` man page is the primary tool used to install and administer non-global zones. Operations using the `zoneadm` command must be run from the global zone on the target system.

In addition to unpacking files from the archive, the install process performs checks, required postprocessing, and other functions to ensure that the zone is optimized to run on the host.

If you created an Oracle Solaris system archive from an existing system and use the `-p` (preserve `sysidcfg`) option when you install the zone, then the zone will have the same identity as the system used to create the image.

If you use the `-u` (`sys-unconfig`) option when you install the target zone, the zone produced will not have a hostname or name service configured.

Caution – You *must* use either the `-p` option or the `-u` option. If you do not specify one of these two options, an error results.

Installer Options

Option	Description
`-a`	Location of archive from which to copy system image. Full flash archive and `pax`, `cpio`, `gzip` compressed `cpio`, `bzip` compressed `cpio`, and level 0 `ufsdump` are supported.
`-d` *path*	Location of directory from which to copy system image.
`-d` –	Use the `-d` option with the dash parameter to direct that the existing directory layout be used in the `zonepath`. Thus, if the administrator manually sets up the `zonepath` directory before the installation, the `-d` – option can be used to indicate that the directory already exists.
`-p`	Preserve system identity. Either the `-p` or the `-u` must be used.
`-s`	Install silently.
`-u`	`sys-unconfig` the zone. Either the `-p` or the `-u` must be used.
`-v`	Verbose output.

The `-a` and `-d` options are mutually exclusive.

How to Install the solaris10 Branded Zone

A configured solaris10 branded zone is installed by using the zoneadm command with the install subcommand.

For information about creating images of Oracle Solaris 10 systems, see "Creating the Image for Directly Migrating Oracle Solaris 10 Systems Into Zones" on page 373. To retain the sysidcfg identity from a system image that you created, without altering the image, use the -p option after the install subcommand. To remove the system identity from a system image that you created, without altering the image, use the -u option. The sys-unconfig occurs to the target zone.

The example procedure shows how to use the -a option with the created archive image of a physical installed Oracle Solaris 10 system.

You must be the global administrator or a user with the appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have equivalent authorizations.**

2. **Install the configured zone s10-zone by using the zoneadm install command with the -a option and the path to the archive:**

   ```
   global# zoneadm -z s10-zone install -a /net/machine_name/s10-system.flar -u
   ```

   You will see various messages as the installation completes. This can take some time.

3. **(Optional) If an error message is displayed and the zone fails to install, use the zoneadm list command and the -c and -v options to get the zone state:**

   ```
   global# zoneadm list -cv

   ID  NAME     STATUS      PATH                          BRAND      IP
    0  global   running     /                             ipkg       shared
    -  s10-zone configured  /zones/s10-zone               solaris10  shared
   ```

   - If the state is listed as configured, make the corrections specified in the message and try the zoneadm install command again.
   - If the state is listed as incomplete, first execute this command:

     ```
     global# zoneadm -z my-zone uninstall
     ```

     Then, make the corrections specified in the message and try the zoneadm install command again.

4. **When the installation completes, use the list subcommand with the -i and -v options to list the installed zones and verify the status.**

   ```
   global# zoneadm list -iv
   ```

You will see a display that is similar to the following:

```
ID  NAME     STATUS       PATH                           BRAND      IP
 0  global   running      /                                   ipkg shared
 -  s10-zone  installed    /zones/s10-zone                 solaris10 shared
```

Example 31–1 `solaris10` Zone Installation

```
# zoneadm -z s10-zone install -u -a /net/machinename/s10_image.flar
      Log File: /var/tmp/s10-zone.install.21207.log
        Source: /net/machinename/s10_image.flar
    Installing: This may take several minutes...
Postprocessing: This may take a minute...

        Result: Installation completed successfully.
      Log File: /zones/s10-zone/root/var/log/s10-zone.install.21207.log
```

Troubleshooting If an installation fails, review the log file. On success, the log file is in `/var/log` inside the zone. On failure, the log file is in `/var/tmp` in the global zone.

If a zone installation is interrupted or fails, the zone is left in the incomplete state. Use the `uninstall` command with the `-F` option to reset the zone to the configured state.

CHAPTER 32

Booting a Zone and Zone Migration

This chapter describes how to boot the installed zone, and also discusses how to migrate the zone to another machine.

If you are booting a zone that does not have the host name or name service configured, read "Internal Zone Configuration" on page 280and "How to Log In to the Zone Console to Perform the Internal Zone Configuration" on page 284 first.

About Booting the `solaris10` Branded Zone

Booting a zone places the zone in the running state. A zone can be booted from the ready state or from the installed state. A zone in the installed state that is booted transparently transitions through the ready state to the running state. Zone login is allowed for zones in the running state.

▼ How to Boot the `solaris10` Branded Zone

You must be the global administrator or a user with the appropriate authorizations in the global zone to perform this procedure.

1. **Be superuser, or have the required rights profile or authorizations.**

2. **Use the `zoneadm` command with the `-z` option, the name of the zone, which is `s10-zone`, and the `boot` subcommand to boot the zone.**

   ```
   global# zoneadm -z s10-zone boot
   ```

3. **When the boot completes, use the `list` subcommand with the `-v` option to verify the status.**

   ```
   global# zoneadm list -v
   ```

You will see a display that is similar to the following:

```
ID  NAME     STATUS       PATH                           BRAND      IP
 0  global   running     /                                ipkg  shared
 1  s10-zone running     /zone/s10-zone                   solaris10  shared
```

See Also For more information on booting zones and boot options, see Chapter 19, "Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks)."

Migrating a `solaris10` Branded Zone to Another Host

A `solaris10` zone can be migrated to another host by using the `zoneadm` command with the `detach` and `attach` subcommands. This process is described in "About Migrating a Zone" on page 294 and "How to Migrate A Non-Global Zone" on page 295.

Note that the `zoneadm attach -a` command takes an archive of a zone, *not* an archive of a physical system.

◆ ◆ ◆ CHAPTER 33

solaris10 Branded Zone Login and Post-Installation Configuration

This chapter discusses logging in to solaris10 zones and using sysidcfg to complete system identification.

solaris10 Branded Internal Zone Configuration

Note that you perform the internal zone configuration when you log in to the unconfigured zone for the first time. This is described in "Internal Zone Configuration" on page 280.

You must accept the network configuration already specified in zonecfg for shared-IP zones.

If you plan to use an /etc/sysidcfg file to perform initial zone configuration, as described in "How to Use an /etc/sysidcfg File to Perform the Initial Zone Configuration" on page 286, create the sysidcfg file and place it the zone's /etc directory before you boot the zone.

▼ How to Log In to the Zone Console to Complete System Identification

You must be the global administrator or a user with the Zone Security rights profile in the global zone to perform this procedure.

1. **Be superuser, or have the appropriate rights profile or authorizations.**

2. **Use the zlogin command with the -C option and the name of the zone, s10-zone in this procedure.**

```
global# zlogin -C s10-zone
```

3. **From another terminal window, boot the zone.**

```
global# zoneadm -z s10-zone boot
```

You will see a display similar to the following in the zlogin window:

```
[NOTICE: Zone booting up]
```

4 The first time you log in to the console, you are prompted to answer a series of questions. Your screen will look similar to this:

```
SunOS Release 5.10 Version Generic_Virtual 64-bit
Copyright 1983-2010 Sun Microsystems, Inc.  All rights reserved
Use is subject to license terms.

Hostname: s10-zone
Select a Language

     0. English
     1. fr
Please make a choice (0 - 1), or press h or ? for help:

Select a Locale

 0. English (C - 7-bit ASCII)
  1. Canada-English (ISO8859-1)
  2. Thai
  3. U.S.A. (en_US.ISO8859-1)
  4. U.S.A. (en_US.ISO8859-15)
  5. Go Back to Previous Screen

Please make a choice (0 - 5), or press h or ? for help:

What type of terminal are you using?
1) ANSI Standard CRT
 2) DEC VT52
 3) DEC VT100
 4) Heathkit 19
 5) Lear Siegler ADM31
 6) PC Console
 7) Sun Command Tool
 8) Sun Workstation
 9) Televideo 910
 10) Televideo 925
 11) Wyse Model 50
 12) X Terminal Emulator (xterms)
 13) Other
Type the number of your choice and press Return:
12
.
.
.
```

For the approximate list of questions you must answer, see "Internal Zone Configuration" on page 280.

5 (Optional) If you are not using two windows as described in step 3, you might have missed the initial prompt for configuration information. If you see the following system message at zone login instead of a prompt:

```
[connected to zone zonename console]
```

Press Return to display the prompt again.

If you enter an incorrect response and try to restart the configuration, you might experience difficulty when you attempt the process again. This occurs because the `sysidtools` can store your previous responses.

If this happens, use the following workaround from the global zone to restart the configuration process.

```
global# zlogin -S zonename /usr/sbin/sys-unconfig
```

For more information on the `sys-unconfig` command, see the `sys-unconfig`(1M) man page.

Glossary

blessed In Perl, the term used to denote class membership of an object.

brand An instance of the BrandZ functionality, which provides non-global zones that contain non-native operating environments used for running applications.

branded zone An isolated environment in which to run non-native applications in non-global zones.

cap A limit that is placed on system resource usage.

capping The process of placing a limit on system resource usage.

data-link An interface at Layer 2 of the OSI protocol stack, which is represented in a system as a STREAMS DLPI (v2) interface. This interface can be plumbed under protocol stacks such as TCP/IP. In the context of Solaris 10 zones, data-links are physical interfaces, aggregations, or VLAN-tagged interfaces . A data-link can also be referred to as a physical interface, for example, when referring to a NIC or a VNIC.

default pool The pool created by the system when pools are enabled.

See also resource pool.

default processor set The processor set created by the system when pools are enabled.

See also processor set.

disjoint A type of set in which the members of the set do not overlap and are not duplicated.

dynamic configuration Information about the disposition of resources within the resource pools framework for a given system at a point in time.

dynamic reconfiguration On SPARC based systems, the ability to reconfigure hardware while the system is running. Also known as DR.

extended accounting A flexible way to record resource consumption on a task basis or process basis in the Solaris operating system.

fair share scheduler A scheduling class, also known as FSS, that allows you to allocate CPU time that is based on shares. Shares define the portion of the system's CPU resources allocated to a project.

FSS See fair share scheduler.

global administrator An administrator with superuser privileges or an equivalent role. When logged in to the global zone, the global administrator or a user granted the appropriate authorizations can monitor and control the system as a whole.

See also zone administrator.

global scope Actions that apply to resource control values for every resource control on the system.

global zone The zone contained on every Oracle Solaris system. When non-global zones are in use, the global zone is both the default zone for the system and the zone used for system-wide administrative control.

See also non-global zone.

heap Process-allocated scratch memory.

local scope Local actions taken on a process that attempts to exceed the control value.

locked memory Memory that cannot be paged.

memory cap enforcement threshold The percentage of physical memory utilization on the system that will trigger cap enforcement by the resource capping daemon.

naming service database In the Projects and Tasks (Overview) chapter of this document, a reference to both LDAP containers and NIS maps.

non-global zone A virtualized operating system environment created within a single instance of the Oracle Solaris operating system. The Oracle Solaris Zones software partitioning technology is used to virtualize operating system services.

non-global zone administrator See zone administrator.

Oracle Solaris 10 Zones A complete runtime environment for Solaris 10 applications executing in a `solaris10` branded zone on a system running the Oracle Solaris 11 Express release.

Oracle Solaris Zones A software partitioning technology used to virtualize operating system services and provide an isolated, secure environment in which to run applications.

page in To read data from a file into physical memory one page at a time.

page out To relocate pages to an area outside of physical memory.

pool See resource pool.

pool daemon The `poold` system daemon that is active when dynamic resource allocation is required.

processor set A disjoint grouping of CPUs. Each processor set can contain zero or more processors. A processor set is represented in the resource pools configuration as a resource element. Also referred to as a pset.

See also disjoint.

project A network-wide administrative identifier for related work.

resident set size The size of the resident set. The resident set is the set of pages that are resident in physical memory.

resource An aspect of the computing system that can be manipulated with the intent to change application behavior.

resource capping daemon A daemon that regulates the consumption of physical memory by processes running in projects that have resource caps defined.

resource consumer Fundamentally, a Solaris process. Process model entities such as the project and the task provide ways of discussing resource consumption in terms of aggregated resource consumption.

resource control A per-process, per-task, or per-project limit on the consumption of a resource.

resource management A functionality that enables you to control how applications use available system resources.

resource partition An exclusive subset of a resource. All of the partitions of a resource sum to represent the total amount of the resource available in a single executing Solaris instance.

resource pool A configuration mechanism that is used to partition machine resources. A resource pool represents an association between groups of resources that can be partitioned.

resource set A process-bindable resource. Most often used to refer to the objects constructed by a kernel subsystem offering some form of partitioning. Examples of resource sets include scheduling classes and processor sets.

RSS See resident set size.

scanner A kernel thread that identifies infrequently used pages. During low memory conditions, the scanner reclaims pages that have not been recently used.

static pools configuration A representation of the way in which an administrator would like a system to be configured with respect to resource pools functionality.

task In resource management, a process collective that represents a set of work over time. Each task is associated with one project.

whole root zone A type of non-global zone in which all of the required system software and any additional packages are installed into the private file systems of the zone.

working set size The size of the working set. The working set is the set of pages that the project workload actively uses during its processing cycle.

workload An aggregation of all processes of an application or group of applications.

WSS See also working set size.

zone administrator The privileges of a zone administrator are confined to a non-global zone.

See also global administrator.

zone state The status of a non-global zone. The zone state is one of configured, incomplete, installed, ready, running, or shutting down.

Index