Introduction to Oracle® Solaris Zones

Part No: E36848-03
December 2014

ORACLE

Contents

Using This Documentation

- **Overview** – Describes the zones technology and the resources available in the zone.
- **Audience** – System administrators, technicians, and authorized service providers.
- **Required knowledge** – Experience with the Oracle Solaris operating system, including knowledge of network configuration and resource allocation.

Product Documentation Library

Late-breaking information and known issues for this product are included in the documentation library at http://www.oracle.com/pls/topic/lookup?ctx=E36784.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPTER 1

Oracle Solaris Zones Introduction

The Oracle™ Solaris Zones feature 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:

The next chapter explains zone configuration resources and properties.

If you are ready to start creating zones on your system, skip to "Creating and Using Oracle Solaris Kernel Zones " and "Creating and Using Oracle Solaris Zones ". If you are ready to migrate a system running Oracle Solaris 10, which can include any non-global `native` zones on that system, into zones on an Oracle Solaris 11 system, see "Creating and using Oracle Solaris 10 Zones ".

Note - For information on using zones on an Oracle Solaris Trusted Extensions system, see Chapter 13, "Managing Zones in Trusted Extensions," in "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. The non-global zone, referred to as a *zone*, is a virtualized operating system environment created within a single instance of the Oracle Solaris operating system. The instance of the operating system is called the global zone. The Oracle Solaris Kernel Zone can run a Support Repository Update (SRU) or kernel version that is different from that of the host.

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

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 root credentials cannot view or affect activity in other zones. Use Oracle Solaris Zones to 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 Oracle Solaris 11 release. The upper limit for the number of `solaris` and `solaris10` 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. These concepts are discussed in Chapter 1, “How to Plan and Configure Non-Global Zones,” in “Creating and Using Oracle Solaris Zones ”.

For more information on these concepts if you are running Oracle Solaris Kernel Zones, see “Hardware and Software Requirements for Oracle Solaris Kernel Zones” in “Creating and Using Oracle Solaris Kernel Zones ”.

Zones Supported on This Release

Non-global `solaris` and `solaris10` brand zones running within a single host global zone are supported on all architectures that the Oracle Solaris 11.2 release has defined as supported platforms.

Oracle Solaris Kernel Zones can run on T4+ and M5+ SPARC machines, Nehalem+ Intel machines, and Barcelona+ AMD machines. For information about kernel zones system requirements, see "Hardware and Software Requirements for Oracle Solaris Kernel Zones" in "Creating and Using Oracle Solaris Kernel Zones ".

Immutable Zones

Immutable Zones are `solaris` zones with read-only roots. Both global and non-global zones can be Immutable Zones. A read-only zone can be configured by setting the `file-mac-profile` property. Several configurations are available. A read-only zone root expands the secure runtime boundary.

Oracle Solaris Immutable Global Zones extended the Immutable Zones feature to the global zone. For Immutable Zones and Immutable Kernel Zones, the Trusted Path login can be invoked through the `zlogin` command `zlogin`(1).

Zones that are given additional datasets using `zonecfg add dataset` still have full control over those datasets. Zones that are given additional file systems using `zonecfg add fs` have full control over those file systems, unless the file systems are set read-only.

See Chapter 12, "Configuring and Administering Immutable Zones," in "Creating and Using Oracle Solaris Zones " for more information.

About Converting `ipkg` Zones to `solaris` Zones

To support Oracle Solaris 11 Express release customers, any zone configured as an `ipkg` zone is converted to a `solaris` zone and reported as `solaris` upon `pkg` update or `zoneadm attach` to Oracle Solaris 11.2. The `ipkg` name will be mapped to the `solaris` name if used when configuring zones. Import of a `zonecfg` file exported from an Oracle Solaris 11 Express host will be supported.

The output of commands such as `zonecfg info` or `zoneadm list -v` displays a brand of `solaris` for default zones on an Oracle Solaris 11.2 system.

Zone Brands in This Release

Oracle Solaris Kernel Zones

The Oracle Solaris Kernel Zones feature provides a full kernel and user environment within a zone, and also increases kernel separation between the host and the zone. The brand name is `solaris-kz`. Kernel zones are managed from the global zone by using the existing tools `zonecfg`, `zoneadm`, and `zlogin`. As the administrator of a kernel zone, you have greater flexibility in configuring and managing the zone than a default `solaris` zone administrator. For example, the you can fully update and modify the zone's installed packages, including the kernel version, without being limited to the packages installed in the global zone. You can manage storage private to the zone, create and destroy ZFS pools, and configure iSCSI and CIFS. You can install `solaris` zones within the kernel zone to produce hierarchical (nested) zones. Kernel zones support suspend and resume. You can migrate a kernel zone by suspending the zone on the source machine and resuming the zone on the target machine.

To use Oracle Solaris Kernel Zones, the package `brand-solaris-kz` must be installed on your system. To determine whether your machine supports kernel zones, see “About Oracle Solaris Zones in This Release” on page 25. You can also run the `virtinfo` command on your machine if Oracle Solaris 11.2 is installed. For more information about Oracle Solaris Kernel Zones, see “Creating and Using Oracle Solaris Kernel Zones ” and the `solaris-kz`(5) man page. For more information about the `virtinfo` command, see “How to Verify Kernel Zone Support on a Host” in “Creating and Using Oracle Solaris Kernel Zones ” and the `virtinfo`(1M)man page.

Default Oracle Solaris Zones

The Oracle Solaris Zones feature 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. Zones use resource management components to control how applications use available system resources. For additional information on resource management features, see “Administering Resource Management in Oracle Solaris 11.2 ”.

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.

For additional isolation, zones with a read-only root, called Immutable Zones, can be configured.

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 operating system. Applications run unmodified in the secure environment provided by the non-global zone. 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 release. These zones are used to migrate Oracle Solaris 10 systems into zones on Solaris 11. A `solaris10` branded zone cannot be an NFS server.

For more information, see “Creating and using Oracle Solaris 10 Zones ”.

Zone Brand Overview

Differences between `solaris-kz` zones and `solaris` and `solaris10` brand zones are shown below.

TABLE 1-1 Comparison of Oracle Solaris Zone Brand Features

Component	`solaris-kz` Brand	`solaris` and `solaris10` Brands
Supported Hardware	Supported on specified hardware. See link to OTN site or HCL.	Supported on all Oracle Solaris 11.2 systems. See HCL.
Memory Management	A fixed amount of physical RAM must be allocated to the `solaris-kz` virtual platform.	Can share the physical RAM allocated to the global zone.
Kernel Version	A kernel zone can run a different kernel version or SRU level than the host.	Kernel version must be the same as that of the global zone.
Storage and Device Management	Performs all storage access. Kernel zones do not support `zpool` or `rootzpool` resources.	Storage can be made available at the file system level though the `fs`, `zpool`, and `dataset` `zonecfg` resources.
Networking	Only exclusive-IP zones are supported.	Exclusive-IP and shared-IP zones are supported.

Using Oracle Solaris Zones on an Oracle Solaris Trusted Extensions System

For information about using zones on an Oracle Solaris Trusted Extensions system, see Chapter 13, “Managing Zones in Trusted Extensions,” in “Trusted Extensions Configuration and Administration ”. Note that only the `labeled` brand can be booted on an Oracle Solaris Trusted Extensions system.

Oracle Solaris Cluster Zone Clusters

Zone clusters are a feature of Oracle Solaris Cluster software. A zone cluster is a group of non-global zones that serve as the nodes of the zone cluster. One non-global zone is created on each global cluster node that is configured with the zone cluster. The nodes of a zone cluster can be of either the `solaris` brand or the `solaris10` brand, and use the cluster attribute. No other brand type is permitted. You can run supported services on the zone cluster in the same way as on a global cluster, with the isolation that is provided by zones. For more information, see the "Oracle Solaris Cluster System Administration Guide ".

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, Oracle Solaris 10 Zones are branded non-global zones 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 release, you must migrate any existing zones to the target system first. Because `solaris10` zones do not nest, the system migration process renders any existing zones unusable. See Chapter 3, “Migrating an Oracle Solaris 10 native Non-Global Zone Into an Oracle Solaris 10 Zone,” in “Creating and using 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.

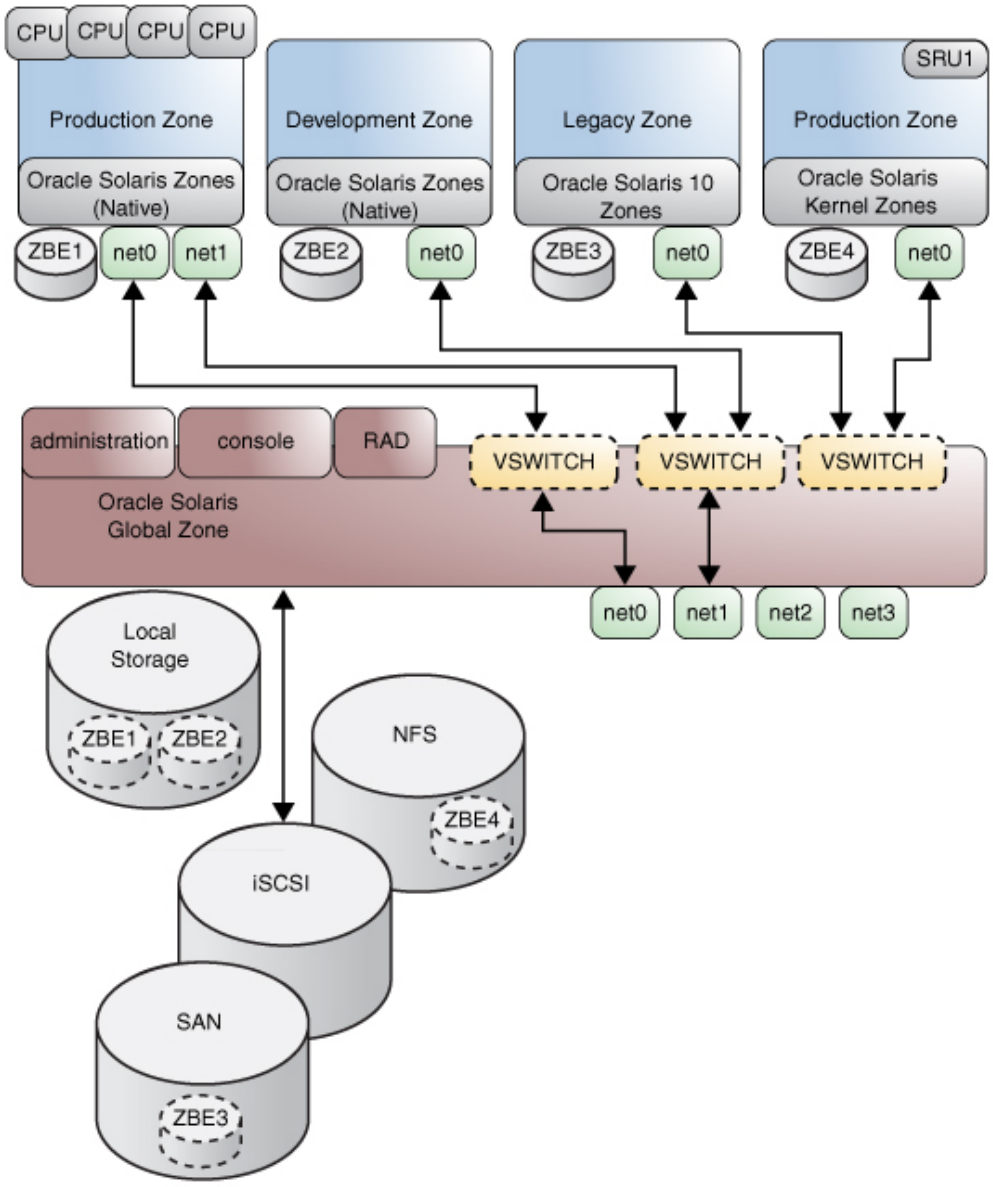
A brand can also provide a plug-in library for `librtld_db`. The plug-in library allows Oracle Solaris tools such as the debugger, described in `mdb`(1), and DTrace, described in `dtrace`(1M), to access the symbol information of processes running inside a branded zone.

Note that zones do not support statically linked binaries.

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.

FIGURE 1-1 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 underutilized 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. 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. Exclusive-IP is the default type. For more information about IP types in zones, see “Zone Network Interfaces” on page 38. For configuration information, see “How to Configure the Zone” in “Creating and Using Oracle Solaris Zones ”.

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 running on a physical system. Appropriately privileged processes running in the global zone can access objects associated with other zones.

In some cases, 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`. if you `zlogin` to a kernel zone, it also reports that it has ID 0, because it is a virtual global zone. Zone names and numeric IDs are discussed in "How to Configure the Zone" in "Creating and Using Oracle Solaris Zones ".

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" in "Creating and Using Oracle Solaris Zones ".

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 51.

The scheduling class for a non-global zone is set to the scheduling class for the system by default. See "Scheduling Class" on page 34 for a discussion of methods used to set the scheduling class in a zone.

Block device multipathing is handled by `scsi_vhci`(7D). The form of the `lu: storage` URI you select for your configuration determines how the configuration is used. For more information about using `lu:` URIs with multipathing, see the `suri`(5) man page.

Summary of Zones by Function

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
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

Type of Zone	Characteristic
	■ 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 administrative rights. 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 32. The privileges of a zone administrator are confined to a specific non-global zone.

How Non-Global Zones Are Created

You can specify the configuration and installation of non-global zones as part of an Automated Install (AI) client installation. See “Installing Oracle Solaris 11.2 Systems ” for more information. Oracle Solaris Kernel Zones primarily are created using the direct installation method. Kernel zone creation methods are documented in “Installing a Kernel Zone” in “Creating and Using Oracle Solaris Kernel Zones ”.

To create a zone on an Oracle Solaris system, the global administrator 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*.

For information about zone configuration, see Chapter 2, “Non-Global Zone Configuration Overview”. For information about zone installation, see Chapter 2, “About Installing, Shutting

Down, Halting, Uninstalling, and Cloning Non-Global Zones," in "Creating and Using Oracle Solaris Zones ". For information about zone login, see Chapter 4, "About Non-Global Zone Login," in "Creating and Using Oracle Solaris Zones ".

To configure and install Oracle Solaris Kernel Zones, see "Creating and Using Oracle Solaris Kernel Zones ".

Non-Global Zone State Model

A non-global zone can be in one of the following seven 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`.
Unavailable	Indicates that the zone has been installed, but cannot be verified, made ready, booted, attached, or moved. A zone enters the unavailable state at the following times: ■ When the zone's storage is unavailable and `svc:/system/zones:default` begins, such as during system boot ■ When the zone's storage is unavailable ■ When archive-based installations fail after successful archive extraction ■ When the zone's software is incompatible with the global zone's software, such as after an improper `-F` (force) attach
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 3, "Installing, Booting, Shutting Down, Halting, Uninstalling, and Cloning Non-Global Zones," in "Creating and Using Oracle Solaris Zones " and the `zoneadm`(1M) man page describe how to use the `zoneadm` command to initiate transitions between these states.

In addition, Oracle Solaris Kernel Zones have three *auxiliary states*, which are used to notify the host with additional information about the current zone state.

Suspended	Primary state is halted, with an auxiliary state of suspended.
Debugging	The zone is running, but the zone cannot respond to external events, such as networking. `zlogin` checks for this state and waits until the state is cleared before starting a `zlogin` session.
Panicked	The zone has panicked, but the zone cannot respond to external events until it is rebooted.

For additional information, see "Creating and Using Oracle Solaris Kernel Zones " and the `solaris-kz`(5) man page.

TABLE 1-2 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`
	
	`zoneadm -z` *zonename* `mark` *incomplete*
	
	`zoneadm -z` *zonename* `mark` *unavailable*
	
	You can use the `zonecfg` command to rename a zone in the configured state. Note that you can use the `zoneadm` command to rename an Oracle Solaris Zone or Oracle Solaris 10 Zone in either the configured or installed state.

Current Zone State	Applicable Commands
Incomplete	`zoneadm -z` *zonename* `uninstall`
Unavailable	`zoneadm -z` *zonename* `uninstall` uninstalls the zone from the specified system. `zoneadm -z` *zonename* `attach` `zonecfg -z` *zonename* can be used to change `zonepath` and any other property or resource that cannot be changed when in the installed state.

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.
	
	You can use the `zoneadm` command to rename an Oracle Solaris Zone or Oracle Solaris 10 Zone in the configured or installed state.
	
	`zoneadm -z` *zonename* `mark` *incomplete*
	
	`zoneadm -z` *zonename* `mark` *unavailable*

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.

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.
	
	`zoneadm -z shutdown` cleanly shuts down the zone.
	
	`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`, `anet`, `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.

Zones-Related SMF Services

Zones-related SMF services in the global zone include the following:

`svc:/system/zones:default`	Starts each zone that has `autoboot=true`.
`svc:/system/zones-install:default`	Performs zone installation on first boot, if needed.
`svc:/application/pkg/zones-proxyd:default`	Used by the packaging system to provide zones access to the system repository.
`svc:/application/pkg/system-repository:default`	Caching proxy server that caches `pkg` data and metadata used during zone installation and other `pkg` operations. See the `pkg`(1) and `pkg`(5) man pages.
`svc:/system/zones-monitoring:default`	Controls `zonestatd`.

The `svc:/application/pkg/zones-proxy-client:default` zones proxy client SMF service runs only in the non-global zone. The service is used by the packaging system to provide zones access to the system repository.

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" in "Creating and Using Oracle Solaris Zones ". The `zonestat` utility also reports on network bandwidth utilization in exclusive-IP zones. An exclusive-IP zone has its own IP-related state and one or more dedicated data-links.

The `fsstat` utility can be used to report file operations statistics for non-global zones. See the `fsstat`(1M) man page and "Monitoring Non-Global Zones Using the fsstat Utility" in "Creating and Using Oracle Solaris Zones ".

Capabilities 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. The applications are also prevented from monitoring or intercepting each other's network traffic, file system data, or process activity.
Network Isolation	Zones are configured as exclusive-IP type by default. The zones are isolated from the global zone and from each other at the IP layer. This isolation is useful for both operational and security reasons. Zones can be used to consolidate applications that must communicate on different subnets using their own LANs or VLANs. Each zone can also define its own IP layer security rules.
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 23 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.

About Oracle Solaris Zones in This Release

This section provides an overview of Oracle Solaris Zones features, including Oracle Solaris Kernel Zones.

The default non-global zone in this release is `solaris`, described in this guide and in the `solaris`(5) man page.

To verify the Oracle Solaris release and the machine architecture, type:

```
#uname -r -m
```

The virtinfo command described in the `virtinfo`(1M) man page is used to obtain the following information:

- Determine system support for Oracle Solaris virtualization technologies
- Detect the type of virtual environment Oracle Solaris is running in, such as Oracle VM Server for SPARC

The `solaris` 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 a `solaris` brand non-global zone. 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 `solaris` zones use this model.

Default `ipkg` zones created on the Oracle Solaris 11 Express release will be mapped to `solaris` zones. See “About Converting `ipkg` Zones to `solaris` Zones” on page 11.

Each non-global zone specified in the Automated Install (AI) manifest is installed and configured as part of a client installation. Non-global zones are installed and configured on the first reboot after the global zone is installed. When the system first boots, the zones self-assembly SMF service, `svc:/system/zones-install:default`, configures and installs each non-global zone defined in the global zone AI manifest. See “Adding and Updating Software in Oracle Solaris 11.2 ” for more information. It is also possible to manually configure and install zones on an installed Oracle Solaris system.

For package updates, persistent proxies should be set in an image by using the `--proxy` option. If a persistent image proxy configuration is not used, `http_proxy` and `https_proxy` environment variables can be set.

Zones can be configured to be updated in parallel instead of serially. The parallel update provides a significant improvement in the time required to update all the zones on a system.

By default, zones are created with the exclusive-IP type. Through the `anet` resource, a VNIC is automatically included in the zone configuration if networking configuration is not specified. For more information, see “Zone Network Interfaces” on page 38.

For information on the `auto-mac-address` used to obtain a `mac-address` for a zone, see the entry `anet` in “Resource Type Properties” on page 60.

A `solaris` zone on shared storage has a `zonecfg rootzpool` resource. A zone is encapsulated into a dedicated zpool. Zones on shared storage access and manage shared storage resources

for zones. Kernel zones do not have `zpool` or `rootzpool` resources. A `solaris` brand zone can use the following shared storage for zone `device` resources, and for `zpool` and `rootzpool` resources.

- iSCSI
- FC LUNs
- DAS

Properties used to specify IP over InfiniBand (IPoIB) data-links are available for the `zonecfg anet` resource. IPoIB is supported for `solaris` and `solaris10` brand zones.

The Reliable Datagram Sockets (RDS) IPC protocol is supported in both exclusive-IP and shared-IP non-global zones.

The `fsstat` utility has been extended to support zones. The `fsstat` utility provides per-zone and aggregate statistics.

`solaris` zones can be NFS servers, as described in the section "Running an NFS Server in a Zone" in "Creating and Using Oracle Solaris Zones ".

Trial-run, also called dry-run, `zoneadm attach -n`, provides `zonecfg` validation, but does not perform package contents validation.

All `zoneadm` options that take files as arguments require absolute paths.

Oracle Solaris 10 Zones provide an Oracle Solaris 10 environment on Oracle Solaris 11. You can migrate an Oracle Solaris 10 system or zone into a `solaris10` zone on an Oracle Solaris 11 system. See "Creating and using Oracle Solaris 10 Zones ".

The `zonep2vchk` tool identifies issues, including networking issues, that could affect the migration of an Oracle Solaris 11 system or an Oracle Solaris 10 system into a zone on a system running the Oracle Solaris 11 release. The `zonep2vchk` tool is executed on the source system before migration begins. The tool also outputs a `zonecfg` script for use on the target system. The script creates a zone that matches the source system's configuration. For more information, see Chapter 7, "About Zone Migrations and the zonep2vchk Tool," in "Creating and Using Oracle Solaris Zones ".

The following differences between `solaris` zones and `native` zones on the Oracle Solaris 10 release should be noted:

- The `solaris` brand is created on Oracle Solaris 11 systems instead of the `native` brand, which is the default on Oracle Solaris 10 systems.
- `solaris` 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 does not use this system. A read-only root zone configuration that is similar to the sparse root type is available.

- 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 capability.
  - The non-global zone root is a ZFS™ dataset.

    A package installed in the global zone is no longer installed into all current and future zones. In general, the global zone's package contents no longer dictate each zone's package contents, for both IPS and SVR4 packaging.
- Non-global zones use boot environments. Zones are integrated with `beadm`, the user interface command for managing ZFS Boot Environments (BEs).

  The `beadm` command is supported inside zones for `pkg` update, just as in the global zone. The `beadm` command can delete any inactive zones BE associated with the zone. See the `beadm`(1M) man page.
- All enabled IPS package repositories must be accessible while installing a zone. See "How to Install a Configured Zone" in "Creating and Using Oracle Solaris Zones " for more information.
- Zone software is minimized to start. Any additional packages the zone requires must be added. See "Adding and Updating Software in Oracle Solaris 11.2 " for more information.

Zones can use Oracle Solaris products and features such as the following:

- Oracle Solaris ZFS encryption
- Network virtualization and QoS
- CIFS and NFS

The following functions cannot be configured in a `solaris-kz` brand zone:

- FC services
- FCoE services

Live Zone Reconfiguration

Use Live Zone Reconfiguration to reconfigure or report on the live configuration of running `solaris` or `solaris10` zones without rebooting. Changes can be made on a temporary or persistent basis.

Use Live Zone Reconfiguration to report live configuration information for `solaris-kz` brand zones.

For more information, see "Creating and Using Oracle Solaris Zones ".

♦ ♦ ♦ CHAPTER 2

Non-Global Zone Configuration Overview

This chapter provides an introduction to non-global zone configuration.

The topics covered in this chapter include the following:

After you have learned about zone configuration, go to Chapter 1, "How to Plan and Configure Non-Global Zones," in "Creating and Using Oracle Solaris Zones " 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.

Using Rights Profiles and Roles in Zone Administration

For information about profiles and roles, see "Protecting and Isolating Applications" in "Oracle Solaris 11 Security Guidelines ".

zonecfg template Property

Use the zonecfg template property to define whether, and how, properties are changed in the following cases:

- When new resource instances are added to a configuration.
- During configuration cloning, when some properties must have unique values. use tokens in the template property to provide these unique values.

TABLE 2-1 zonecfg template Tokens

Token	Description	Usage
%*zonename*	The name of the zone.	Can be used in the brand's metadata, and in zonecfg as input from user or from a template value.
%*network-id*	A unique instance number for network resources net and anet. This number is unique for net and anet resource within the global zone.	Can be used in the brand's metadata,as the default attribute for the id property net and anet resources.
%*resource-id*	A unique instance number within a given resource global scope, within the global zone, for all resource except net and anet.	Can be used in the brand's metadata as the default attribute for the id property.
%*id*	A unique instance number that is the resource's *id* property value.	Can be used in zonecfg as input from user,or from template value. Should be used within a resource scope that supports the id property.
%%	Evaluates to %.	Can be used in the brand's metadata, and in zonecfg as input from user .

Zones remote administration daemon (RAD) module configuration provides a systemic way to express, enforce, or implement changes by using the property templates. See the zonemgr(3RAD) man page. If the rad-zonemgr package was not initially installed on your system and you installed it later using pkg install, you must restart rad:local. Also restart rad:remote, if that was running. To restart, use svcadm(1M). Make sure the RAD daemon loaded the module.

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.
- Ensures that the configuration is free from conflicts. For example, if you have an `anet` resource, the zone is an exclusive-IP type and cannot be a shared-IP zone. Also, the `zonecfg` command issues a warning if an aliased dataset has a potential conflict with devices.

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 55.

Zone Name and Path

You must choose a name for your zone. If you do not specify the path, the default value of `zonepath` is `/system/zones/`*zonename*.

If you 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.

If the path is not specified, the default value of `zonepath` is `/system/zones/`*zonename*.

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.

`file-mac-profile` Property for Immutable Zones

In `solaris` zones, the `file-mac-profile` is used to configure Immutable Zones with read-only roots.

For more information, see Chapter 12, “Configuring and Administering Immutable Zones,” in “Creating and Using Oracle Solaris Zones ”.

`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 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*.

For more information on authorizations, see `auths`(1), `auth_attr`(4), and `user_attr`(4).

`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" in "Administering Resource Management in Oracle Solaris 11.2 ".

`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" in "Administering Resource Management in Oracle Solaris 11.2 ".

The following properties are used to set persistent `dedicated-cpu` resources for `cpus`, `cores` and `sockets`.

`cpus` — Assign specific CPUs to a zone persistently.

`cores` — Assign specific cores to zone persistently.

`sockets` — Assign specified number of sockets persistently.

Note - The `capped-cpu` resource and the `dedicated-cpu` resource are incompatible. The `cpu-shares` rctl and the `dedicated-cpu` resource are incompatible.

Note - Applications that auto-size and automatically scale to the number of available CPUs might not recognize a capped-cpu restriction. Seeing all CPUs as available can adversely affect scaling and performance in applications such as the Oracle database and Java virtual machines (JVM). It can appear that the application is not working or usable. The JVM should not be used with capped-cpu if performance is critical. Applications in affected categories can use the dedicated-cpu resource.

`solaris-kz` Only: `virtual-cpu` Resource

Use the `virtual-cpu` resource to set the number of kernel zone CPUs. The host CPUs dedicated to the kernel zone are defined by the `ncpus` value. The default kernel zone

configuration has 1 CPU. You can add more CPUs to the kernel zone by adding the `virtual-cpu` property.

Note that if the `dedicated-cpu` resource is already defined, the default number of virtual CPUs configured in the virtual platform matches the lower value of the `ncpus` range in the `dedicated-cpu` resource. Setting both the `dedicated-cpu` and the `virtual-cpu` resources is not necessary.

`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.

Note - Applications that auto-size and automatically scale to the number of available CPUs might not recognize a `capped-cpu` restriction. Seeing all CPUs as available can adversely affect scaling and performance in applications such as the Oracle database and Java virtual machines (JVM). It can appear that the application is not working or usable. The JVM should not be used with `capped-cpu` if performance is critical. Applications in affected categories can use the `dedicated-cpu` resource. See “`dedicated-cpu` Resource” on page 32.

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" in "Administering Resource Management in Oracle Solaris 11.2 " and "How to Associate a Pool With a Scheduling Class" in "Administering Resource Management in Oracle Solaris 11.2 ".
- 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. To use the `capped-memory` resource, the `resource-cap` package must be installed in the global zone. Also see "`capped-cpu` Resource" on page 34.

- 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, “About Physical Memory Control Using the Resource Capping Daemon,” in “Administering Resource Management in Oracle Solaris 11.2 ”, Chapter 11, “Administering the Resource Capping Daemon Tasks,” in “Administering Resource Management in Oracle Solaris 11.2 ”, and “How to Configure the Zone” in “Creating and Using Oracle Solaris Zones ”. To temporarily set a resource cap for a zone, see “How to Specify a Temporary Resource Cap for a Zone” in “Administering Resource Management in Oracle Solaris 11.2 ”.

`solaris` and `solaris10` Only: `rootzpool` Resource

The optional `rootzpool` resource in the `zonecfg` utility is used to create a dedicated zpool for zone installation for `solaris` and `solaris10` brand zones. The zone root zpool can be hosted on shared storage devices defined by one or more Universal Resource Identifiers (URIs). The required `storage` property identifies the storage object URI to contain the root `zfs` file system for a zone. Only one `rootzpool` can be defined for a given zone. The storage is automatically configured for the zone when the zone is booted.

The corresponding zpools are automatically created or imported during zone installation or zone attach operations. For both the rootzpool and zpool resources, you can automatically create zpool mirrors as soon as the zone is installed. For more information, see Chapter 14, “Getting Started With Oracle Solaris Zones on Shared Storage,” in “Creating and Using Oracle Solaris Zones ”.

When the zone is uninstalled or detached, the following actions take place:

- The corresponding zpools are automatically exported or destroyed.
- The storage resources are automatically unconfigured.

To reuse a pre-created zpool for a zone installation, the zpool must be exported from the system.

The zones framework supports the following URI types:

- `dev`

  Local device path URI

  Format:

  `dev:`*local-path-under-*`/dev`
  `dev://`*absolute-path-with-dev*
  `dev:`*absolute-path-with-dev*

  Examples:

  ```
  dev:dsk/c7t0d0s0
  dev:///dev/dsk/c7t0d0s0
  dev:/dev/dsk/c7t0d0s0
  dev:chassis/SYS/HD1/disk
  ```

- `lu` (Logical Unit)

  Fibre Channel (FC) and Serial Attached SCSI (SAS)

  Format:

  ```
  lu:luname.naa.ID
  lu:luname.eui.ID
  lu:initiator.naa.ID,target.naa.ID,luname.naa.ID
  lu:initiator.naa.ID,target.naa.ID,luname.eui.ID
  ```

  Examples:

  ```
  lu:luname.naa.5000c5000288fa25
  lu:luname.eui.0021280001cf80f6
  lu:initiator.naa.2100001d38089fb0,target.naa.2100001d38089fb0,luname.naa.5000c5000288fa25
  lu:initiator.naa.2100001d38089fb0,target.naa.2100001d38089fb0,luname.eui.0021280001cf80f6
  ```

- `iscsi`

  iSCSI URI

  Format:

  ```
  iscsi:///luname.naa.ID
  iscsi:///luname.eui.ID
  iscsi://host[:port]/luname.naa.ID
  iscsi://host[:port]/luname.eui.ID
  iscsi:///target.IQN,lun.LUN
  iscsi://host[:port]/target.IQN,lun.LUN
  ```

  Examples:

  ```
  iscsi:///luname.eui.0021280001cf80f6
  iscsi:///luname.naa.600144f03d70c80000004ea57da10001
  iscsi://[::1]/luname.naa.600144f03d70c80000004ea57da10001
  iscsi://127.0.0.1/luname.naa.600144f03d70c80000004ea57da10001
  iscsi://hostname:1234/luname.eui.0021280001cf80f6
  iscsi://hostname:3260/luname.naa.600144f03d70c80000004ea57da10001

  iscsi://127.0.0.1/target.iqn.com.sun:02:d0f2d311-f703,lun.0
  iscsi:///target.iqn.com.sun:02:d0f2d311-f703,lun.6
  iscsi://[::1]:1234/target.iqn.com.sun:02:d0f2d311-f703,lun.2
  iscsi://hostname:1234/target.iqn.com.sun:4db41b76-e3d7-cd2f-bf2d-9abef784d76c,lun.0
  ```

The `suriadm` tool is used to administer shared objects based on storage URIs. For information about IDs, the Name Address Authority (NAA), and obtaining URIs for existing storage objects, see the `suriadm`(1M) and `suri`(5) man pages.

The system names the newly created or imported `rootzpool` for its associated zone. The assigned name has the form *zonename*`_rpool`.

The `storage` property is managed using the following commands from inside the `rootzpool` resource scope:

- `add storage` *URI string*
- `remove storage` *URI string*

Adding a `zpool` Resource Automatically

A `zpool` can be delegated to a non-global zone by configuring the optional `zpool` resource in the `zonecfg` utility. The `zpool` is automatically configured for the zone when it is booted.

The corresponding zpools are automatically created or imported during zone installation or zone attach operations.

When the zone is uninstalled or detached, the following actions take place:

- The corresponding zpools are automatically exported or destroyed.
- The storage resources are automatically unconfigured.

The required `storage` property identifies the storage object URI associated with this resource.

The `storage` property is managed using the following settings in the `zpool` resource scope:

- `add storage` *URI string*
- `remove storage` *URI string*

The `name` property is mandatory for the `zpool` resource. The property is used in the name for a zpool delegated to the zone. The ZFS file system `name` component cannot contain a forward slash (`/`).

The assigned name of the newly created or imported `zpool` is the value of the `name` property. This is the zpool name visible inside the non-global zone. The assigned name of the newly created or imported `zpool` name has the form *zonename_name* when displayed from the global zone.

Note - A zone installation can fail when a storage object contains preexisting partitions, zpools, or UFS file systems. For more information, see Step 4 in "How to Install a Configured Zone" in "Creating and Using Oracle Solaris Zones ".

Zone Network Interfaces

Zone network interfaces configured by the `zonecfg` utility to provide network connectivity are automatically 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. Exclusive IP is the default IP type. A shared-IP zone shares a network interface with the global zone. Configuration in the global zone must be done by the `ipadm` utility to use shared-IP zones. An exclusive-IP zone must have a dedicated network interface. If the exclusive-IP zone is configured using the `anet` resource, a dedicated VNIC is automatically created and assigned to that zone. By using the automated `anet` resource, the requirement to create and configure data-links in the global zone and assign the data-links to non-global zones is eliminated. Use the `anet` resource to accomplish the following:

- Allow the global zone administrator to choose specific names for the data-links assigned to non-global zones
- Allow multiple zones to use data-links of the same name

For backward compatibility, preconfigured data-links can be assigned to non-global zones.

For information about IP features in each type, see "Networking in Exclusive-IP Non-Global Zones" in "Creating and Using Oracle Solaris Zones " and "Networking in Shared-IP Non-Global Zones" in "Creating and Using Oracle Solaris Zones ".

Note - The link protection described in "Securing the Network in Oracle Solaris 11.2 " can be used on a system running zones. This functionality is configured in the global zone.

About Data-Links

A data-link is a physical 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. A data-link is also referred to as a physical interface, for example, a Network Interface Card (NIC). The data-link is the `physical` property configured by using `zonecfg`(1M). The `physical` property can be a VNIC.

By default in Oracle Solaris 11, physical network device names use generic names, such as `net0`, instead of device driver names, such as `nxge0`.

For information about using IP over Infiniband (IPoIB) for `solaris` zones, see the `anet` description in "Resource Type Properties" on page 60.

About Elastic Virtual Switch and Zones

For an `anet` resource that connects to an Elastic Virtual Switch (EVS) with the `evs` and `vport` properties set, the properties of that `anet` resource are encapsulated in the `evs` and `vport` pair. You cannot change any of the following properties for an EVS `anet` resource:

- `mac-address`
- `mtu`
- `maxbw`
- `priority`
- `allowed-address`
- `vlan-id`
- `defrouter`
- `lower-link`

The only properties that you can set for an EVS `anet` resource are the following:

- linkname
- `evs`
- `vport`
- `configure-allowed-address`

You must also set the `tenant` resource. Tenants are used for namespace management. The EVS resources defined within a `tenant` are not visible outside that tenant's namespace.

The following input for a zone named *evszone* sets the `tenant` resource for a tenant named *tenantA*. The `zonecfg` `anet` resource properties create a VNIC for a zone that has an `anet` resource that connects to an EVS named *evsa* and a VPort named *vport0*:

```
zonecfg:evszone> set tenant=tenantA

zonecfg:evszone> add anet

zonecfg:evszone> set evs=EVSA

zonecfg:evszone> set vport=vport0
```

For more information, see Chapter 5, “About Elastic Virtual Switches,” in “Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ”.

Shared-IP Non-Global Zones

A shared-IP zone uses an existing IP interface from the global zone. 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 `net` resource of 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.

To use the shared-IP type networking configuration in the global zone, you must use `ipadm`, not automatic network configuration. To determine whether networking configuration is being done by `ipadm`, run the following command. The response displayed must be `DefaultFixed`.

```
# svcprop -p netcfg/active_ncp svc:/network/physical:default
DefaultFixed
```

The IP addresses assigned to shared-IP zones are associated with logical network interfaces.

The `ipadm` command can be used from the global zone to assign or remove logical interfaces in a running zone.

To add interfaces, use the following command:

```
global# ipadm set-addrprop -p zone=my-zone net0/addr1
```

To remove interfaces, use one of the following commands:

```
global# ipadm set-addrprop -p zone=global net0/addr
```

or:

```
global# ipadm reset-addrprop -p zone net0/addr1
```

For more information, see “Shared-IP Network Interfaces” in “Creating and Using Oracle Solaris Zones ”.

Exclusive-IP Non-Global Zones

Exclusive-IP is the default networking configuration for non-global zones.

An exclusive-IP zone has its own IP-related state and one or more dedicated data-links.

The following features can be used 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
- `ipadm` for setting TCP/UDP/SCTP as well as IP/ARP-level tunables
- IP security (IPsec) and Internet Key Exchange (IKE), which automates the provision of authenticated keying material for IPsec security association

There are two ways to configure exclusive-IP zones:

- Use the `anet` resource of the `zonecfg` utility to automatically create a temporary VNIC for the zone when the zone boots and delete it when the zone halts.
- Preconfigure the data-link in the global zone and assigned it to the exclusive-IP zone by using the `net` resource of the `zonecfg` utility. The data-link is specified by using the `physical` property of the `net` resource. The `physical` property can be a VNIC. The `address` property of the `net` resource is not set.

By default, an exclusive-IP zone can configure and use any IP address on the associated interface. Optionally, a comma-separated list of IP addresses can be specified using the `allowed-address` property. The exclusive-IP zone cannot use IP addresses that are not in the `allowed-address` list. Moreover, all the addresses in the `allowed-address` list will automatically be persistently configured for the exclusive-IP zone when the zone is booted. If this interface configuration is not wanted, then the `configure-allowed-address` property must be set to `false`. The default value is `true`.

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” in “Creating and Using Oracle Solaris Zones ” for usage examples.

Inside a running exclusive-IP zone that is assigned its own set of data-links, the `ipadm` 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 in the global zone, by using the `sysconfig` interface described in the `sysconfig`(1M) man page.

The IP configuration of an exclusive-IP zone can only be viewed from the global zone by using the `zlogin` command.

```
global# zlogin zone1 ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
nge0/v4           dhcp     ok           10.134.62.47/24
lo0/v6            static   ok           ::1/128
nge0/_a           addrconf ok           fe80::2e0:81ff:fe5d:c630/10
```

Reliable Datagram Sockets Support in Non-Global Zones

The Reliable Datagram Sockets (RDS) IPC protocol is supported in both exclusive-IP and shared-IP non-global zones. The RDSv3 driver is enabled as SMF service `rds`. By default, the service is disabled after installation. The service can be enabled within a given non-global zone by a zone administrator granted appropriate authorizations. After `zlogin`, `rds` can be enabled in each zone in which it is to run.

EXAMPLE 2-1 How to Enable the `rds` Service in a Non-Global Zone

1. To enable RDSv3 service in an exclusive-IP or shared-IP zone, zlogin and execute the `svcadm enable` command:

```
# svcadm enable rds
```

2. Verify that `rds` is enabled:

```
# svcs rds
    STATE          STIME    FMRI
    online         22:50:53 svc:/system/rds:default
```

For more information, see the `svcadm`(1M) man page.

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. IP address spoofing can be disabled by setting the `allowed-address` property. For the `anet` resource, additional protections such as `mac-nospoof` and `dhcp-nospoof` can be enabled by setting the `link-protection` property.

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

Each zone has a ZFS dataset delegated to it by default. This default delegated dataset mimics the dataset layout of the default global zone dataset layout. A dataset called .../rpool/ROOT contains boot environments. This dataset should not be manipulated directly. The rpool dataset, which must exist, is mounted by default at .../rpool. The .../rpool/export, and .../rpool/export/home datasets are mounted at /export and /export/home. These non-global zone datasets have the same uses as the corresponding global zone datasets, and can be managed in the same way. The zone administrator can create additional datasets within the .../rpool, .../rpool/export, and .../rpool/export/home datasets.

You should *not* use the zfs command described in the zfs(1M) man page to create, delete, or rename file systems within the hierarchy that starts at the zone's rpool/ROOT file system. The zfs command can be used to set properties other than canmount, mountpoint, sharesmb, zoned, com.oracle.*:*, com.sun:*, and org.opensolaris.*.*..

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

These sets can include, for example, the following file systems:

- ZFS file systems with a mountpoint other than none or legacy that also have a value of yes for the canmount property.
- File systems specified in a zone's /etc/vfstab file.
- AutoFS and AutoFS-triggered mounts. autofs properties are set by using the sharectl described in sharectl(1M).
- 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 file systems within a zone's default delegated dataset, 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” in “Creating and Using Oracle Solaris Zones ” for more information.

For more information about datasets, see the datasets(5) man page. For more information about BEs, see "Creating and Administering Oracle Solaris 11.2 Boot Environments ".

File System Mounts and Updating

It is not supported to mount a file system in a way that hides any file, symbolic link, or directory that is part of the zone's system image as described in the pkg(5) man page. For example, if there are no packages installed that deliver content into `/usr/local`, it is permissible to mount a file system at `/usr/local`. However, if any package, including legacy SVR4 packages, delivers a file, directory, or symbolic link into a path that begins with `/usr/local`, it is not supported to mount a file system at `/usr/local`. It is supported to temporarily mount a file system at `/mnt`.

Due to the order in which file systems are mounted in a zone, it is not possible to have an `fs` resource mount a file system at `/export/filesys` if `/export` comes from the zone's `rpool/export` dataset or another delegated dataset.

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 Types and Properties" on page 55 for more information.

`/dev` File System in Non-Global 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" in "Creating and Using Oracle Solaris Zones ".

Removable `lofi` Device in Non-Global Zones

A removable loopback file `lofi` device, which works like a CD-ROM device, can be configured in a non-global zone. You can change the file that the device maps to and create multiple `lofi` devices to use the same file in readonly mode. This type of `lofi` device is created by using the `lofiadm` command with the `-r` option. A file name is not required at creation time.

During the lifecycle of a removable `lofi` device, a file can be associated with an empty device, or dissociated from a device that is not empty. A file can be associated with multiple removable `lofi` devices safely at the same time. A removable `lofi` device is read-only. You cannot remap a file that has been mapped to either a normal read-write `lofi` device or to a removable `lofi` device. The number of potential `lofi` devices is limited by the `zone.max-lofi` resource control, which can be set by using `zonecfg`(1M) in the global zone.

Once created, a removable `lofi` device is read-only. The `lofi` driver will return an error on any write operation to a removable `lofi` device.

The `lofiadm` command is also used to list removable `lofi` devices.

EXAMPLE 2-2 Create a Removable `lofi` Device With an Associated File

```
# lofiadm -r /path/to/file
/dev/lofi/1
```

EXAMPLE 2-3 Create an Empty Removable `lofi` Device

```
# lofiadm -r
/dev/lofi/2
```

EXAMPLE 2-4 Insert a File Into a Removable `lofi` Device

```
# lofiadm -r /path/to/file /dev/lofi/1
/dev/lofi/1
```

For more information, see the `lofiadm`(1M), `zonecfg`(1M), and `lofi`(7D) man pages. Also see Table 2-2.

Disk Format Support in Non-Global Zones

Disk partitioning and use of the `uscsi` command are enabled through the `zonecfg` tool. See `device` in "Resource Type Properties" on page 60 for an example. For more information on the `uscsi` command, see `uscsi`(7I).

- Delegation is only supported for `solaris` zones.
- Disks must use the `sd` target as shown by using the `prtconf` command with the `-D` option. See `prtconf`(1M).

Kernel Zones Device Resources With Storage URIs

The following support is available:

- `solaris-kz` supports the `bootpri` and `id` properties in device resources.
  - Only set `bootpri` on disks that will be part of the root pool for the zone. If you set bootpri on disks that will **not** be part of the root pool for the zone, you could damage the data on the disk.
  - `id` controls the instance of the disk in the kernel zone. for example, `id=5` means that the disk will be `c1d5` in the zone.
- The root zpool that is created on bootable `solaris-kz` disks can be imported into the global zone during installation,. At this time, the root zpool is visible with the `zpool` command. See `zpool`(1M) for more information.

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" in "Creating and Using Oracle Solaris Zones ", "How to Configure the Zone" in "Creating and Using Oracle Solaris Zones ", and `privileges`(5).

Resource Pool Association

If you have configured resource pools on your system as described in Chapter 13, "Creating and Administering Resource Pools Tasks," in "Administering Resource Management in Oracle

Solaris 11.2 ", you can use the `pool` property to associate the zone with one of the resource pools when you configure the zone.

You can 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. You can use `dedicated-cpu` properties to assign CPUs, cores, and sockets to a zone. The system dynamically creates a temporary pool for use while the zone is running. With specification through `zonecfg`, pool settings propagate during migrations. If you are configuring Oracle Solaris Kernel Zones, also see the `virtual-cpu` resource.

The `pool` property can be used to configure multiple zones that share the same pool.

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.

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" in "Creating and Using Oracle Solaris Zones ".

The preferred, simpler method for setting a zone-wide resource control is to use the property name or resource, such as `capped-cpu`, instead of the `rctl` resource, such as `cpu-cap`.

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 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.cpu-cap` 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" in "Creating and Using Oracle Solaris Zones ". 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 "Available Resource Controls" in "Administering Resource Management in Oracle Solaris 11.2 " 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 "Available Resource Controls" in "Administering Resource Management in Oracle Solaris 11.2 " 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" in "Administering Resource Management in Oracle Solaris 11.2 ". 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 2-2 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)	
`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)	

Control Name	Global Property Name	Description	Default Unit	Value Used For
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" in "Creating and Using Oracle Solaris Zones ". Limits specified through the prctl command are not persistent. The limits are only in effect until the system is rebooted.

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" in "Creating and Using Oracle Solaris Zones ".

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 1, “How to Plan and Configure Non-Global Zones,” in “Creating and Using Oracle Solaris Zones ”.

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 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 Oracle Solaris 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" in "Creating and Using Oracle Solaris Zones " 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. Use of the clear command on a property clears the value to the default value of the property.
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. Verify the syntax of any `rootzpool` resource group and its properties. The accessibility of any storage specified by a URI is not verified.

`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 52 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 Types and Properties

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 63 characters.
- The name `global` is reserved for the global zone.
- Names beginning with `SYS` are reserved and cannot be used.

`zonepath` The `zonepath` property specifies the path under which the zone will be installed. 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. The zone path must be owned by `root` with the mode `700`. If the zone path does not exist, it

will be automatically created during installation. If the permissions are incorrect, they will be automatically corrected.

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`. 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 is created automatically when the zone is installed or attached. If a ZFS dataset cannot be created, the zone will not install or attach.

Path	Description
`/zones/my-zone`	`zonecfg zonepath`
`/zones/my-zone/root`	Root of the zone

See “Traversing File Systems” in “Creating and Using Oracle Solaris Zones ” for more information.

In the `zonecfg template` property, the default value of `zonepath` is `/system/zones/`*zonename*.

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” in “Creating and Using Oracle Solaris Zones ” for instructions.

`autoboot`

If this property is set to `true`, the zone is automatically booted when the global zone is booted. It is set to `false` by default. 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” in “Creating and Using Oracle Solaris Zones ” for information on this setting during `pkg` 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*.

`limitpriv`

This property is used to specify a privilege mask other than the default. See "Privileges in a Non-Global Zone" in "Creating and Using Oracle Solaris Zones ".

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 34 for additional information and tips.

`ip-type`

This property is required to be set for all non-global zones. See "Exclusive-IP Non-Global Zones" on page 41, "Shared-IP Non-Global Zones" on page 40, and "How to Configure the Zone" in "Creating and Using Oracle Solaris Zones ".

`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`. `ncores`, `cores`, and `sockets`. For more information, see "`dedicated-cpu` Resource" on page 32.

solaris-kz Only:
`virtual-cpu`

This `solaris-kz` resource dedicates a subset of the system's processors to the zone while it is running. The `virtual-cpu` resource provides limits for `ncpus`. For more information, see "`solaris-kz` Only: `virtual-cpu` Resource" on page 33.

`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 34.

`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. To use the `capped-memory` resource, the `service/resource-cap` package must be installed in the global zone.

`anet`

The `anet` resource automatically creates a temporary VNIC interface for the exclusive-IP zone when the zone boots and deletes it when the zone halts.

`net`

The net resource assigns an existing network interface in the global zone to the non-global zone. The network interface resource is the interface name. Each zone can have network interfaces that are set up when the zone transitions from the installed state to the ready state.

`dataset`

A dataset is a generic term for file system, volume, or snapshot. Adding a ZFS™ dataset resource enables the delegation of storage administration to a non-global zone. If the delegated dataset is a file system, the zone administrator can create and destroy file systems within that dataset, and modify properties of the dataset. The zone administrator can create snapshots, child file systems and volumes, and clones of its descendants. If the delegated dataset is a volume, the zone administrator can set properties and create snapshots. 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

When the `zonecfg template` property is used, if a `rootzpool` resource is not specified, the default zonepath dataset is *rootpool*`/VARSHARE/zones/`*zonename*. The dataset is created by the `svc-zones` service with a mountpoint `/system/zones`The remaining properties are inherited from *rootpool*`/VARSHARE/zones/`,

See Chapter 9, "Oracle Solaris ZFS Advanced Topics," in "Managing ZFS File Systems in Oracle Solaris 11.2 ", "File Systems and Non-Global Zones" in "Creating and Using Oracle Solaris Zones " and the `datasets`(5) man page.

Also see Chapter 13, “Troubleshooting Miscellaneous Oracle Solaris Zones Problems,” in “Creating and Using Oracle Solaris Zones ” 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” in “Creating and Using Oracle Solaris Zones ”.

Note - To use UFS file systems in a non-global zone through the `fs` resource, the `system/file-system/ufs` package must be installed into the zone after installation or through the AI manifest script.

The `quota` command documented in `quota`(1M) cannot be used to retrieve quota information for UFS file systems added through the `fs` resource.

`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” in “Creating and Using Oracle Solaris Zones ” and “Device Use in Non-Global Zones” in “Creating and Using Oracle Solaris Zones ”.

`device`

The `zonefigdevice` resource is used to add virtual disks to a non-global zone's platform. 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.

Note - To use UFS file systems in a non-global zone through the `device` resource, the `system/file-system/ufs` package must be installed into the zone after installation or through the AI manifest script.

`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 32.

`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 48 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" in "Creating and Using Oracle Solaris Zones ".

`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.

`solaris` and `solaris10` Only: `rootzpool`

`storage`

Identify the storage object URI to provide a dedicated ZFS zpool for zone installation. For information on URIs and the allowed values for `storage`, see "`solaris` and `solaris10` Only:`rootzpool` Resource" on page 36. During zone installation, the zpool is automatically created, or a pre-created zpool is imported. The name *my-zone*`_rpool` is assigned.

```
zonecfg:my-zone> add rootzpool
zonecfg:my-zone:rootzpool> add storage dev:dsk/c4t1d0
zonecfg:my-zone:rootzpool> end
```

You can add an additional `storage` property if you are creating a mirrored configuration:

```
add storage dev:dsk/c4t1d0
add storage dev:dsk/c4t3d0
```

Only one `rootzpool` resource can be configured for a zone.

`solaris` and `solaris10` Only: `zpool`

`storage`, `name`

Define one or more storage object URIs to delegate a zpool to the zone. For information on URIs and the allowed values for the `storage` property, see "`solaris` and `solaris10` Only:`rootzpool` Resource" on page 36. The allowed values for the `name` property are defined in the `zpool`(1M) man page.

In this example, a `zpool` storage resource is delegated to the zone. The zpool is automatically created, or a previously created zpool is imported during installation. The name of the `zpool` is `my-zone_pool1`.

```
zonecfg:my-zone> add zpool
zonecfg:my-zone:zpool> set name=pool1
zonecfg:my-zone:zpool> add storage dev:dsk/c4t2d0
zonecfg:my-zone:zpool> add storage dev:dsk/c4t4d0
zonecfg:my-zone:zpool> end
```

A zone configuration can have one or more `zpool` resources.

`dedicated-cpu`

`ncpus`, `importance`, `cores`, `cpus`, `sockets`

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
```

Persistently assign cores `0`, `1`, `2`, and `3` to the zone `my-zone`. The following `dedicated-cpu` example uses *cores*, but `cpus=`, `cores=`, and `sockets=` can all be used.

```
zonecfg:my-zone> add dedicated-cpu
zonecfg:my-zone:dedicated-cpu> set cores=0-3
zonecfg:my-zone:dedicated-cpu> end
```

`virtual-cpu` — `ncpus`

Specify the number of CPUs. The following example specifies 3 CPUs for the zone `my-zone`.

```
zonecfg:my-zone> add virtual-cpu
zonecfg:my-zone:dedicated-cpu> set ncpus=3
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
```

To use `capped-memory` resource, the `resource-cap` package must be installed in the global zone.

`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” in “Creating and Using Oracle Solaris Zones ”, “Security Restrictions and File System Behavior” in “Creating and Using Oracle Solaris Zones ”, 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*.

Note - The `quota` command documented in `quota`(1M) cannot be used to retrieve quota information for UFS file systems added through this resource.

`dataset` `name`, `alias`

`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
```

A delegated dataset can have a non-default alias as shown in the following example. Note that a dataset alias cannot contain a forward slash (`/`).

```
zonecfg:my-zone> add dataset
zonecfg:my-zone:dataset> set name=tank/sales
zonecfg:my-zone:dataset> set alias=data
zonecfg:my-zone:dataset> end
```

To revert to the default alias, use `clear alias`.

```
zonecfg:my-zone> clear alias
```

`anet`

`linkname`, `lower-link`, `allowed-address`, `auto-mac-address`, `configure-allowed-address`, `defrouter`, `linkmode` (IPoIB), `mac-address` (non-IPoIB), `mac-slot` (non-IPoIB),`mac-prefix` (non-IPoIB), `mtu`, `maxbw`, `pkey` (IPoIB), `priority`, `vlan-id` (non-IPoIB), `rxfanout`, `rxrings`, `txrings`, `link-protection`, `allowed-dhcp-cids`

`solaris` **Only:** Do not set the following `anet` properties for IPoIB data-links in `zonecfg`.

- `mac-address`
- `mac-prefix`
- `mac-slot`
- `vlan-id`

Do not set the following `anet` properties for non-IPoIB data-links in `zonecfg`.

- `linkmode`
- `pkey`

Set only the following properties for an EVS `anet` resource:

- linkname
- `evs`
- `vport`
- `configure-allowed-address`

The `anet` resource creates an automatic VNIC interface or an IPoIB interface when the zone boots, and deletes the VNIC or IPoIB interface when the zone halts. Note that the `solaris-kz` brand does not support IPoIB. The resource properties are managed through the `zonecfg` command. See the `zonecfg`(1M) man page for the complete text on properties available.

`lower-link` — Specifies the underlying link for the link to be created. When set to `auto`, the `zoneadmd` daemon automatically chooses the link over which the VNIC is created each time the zone boots. You can

	specify any link on which you can create a VNIC as the `lower-link` for an `anet` resource. All IPoIB links are skipped when selecting the data-link for creating the VNIC automatically during boot.
`linkname`	Specify a name for the automatically created VNIC interface or IPoIB interface. Note that `solaris-kz` does not support IPoIB.
`mac-address` **(not for IPoIB)**	Set the VNIC MAC address based on the specified value or keyword. If the value is not a keyword, it is interpreted as a unicast MAC address. See the `zonecfg`(1M) man page for supported keywords. If a random MAC address is selected, the generated address is preserved across zone boots, and zone detach and attach operations. When the default policy `auto-mac-address` is used, Oracle Solaris Zones can obtain a random `mac-address`.
`pkey` **(IPoIB only)**	Set the partition key to be used for creating the IPoIB data-link interface. This property is mandatory. The specified `pkey` is always treated as hexadecimal, whether or not it has the `0x` prefix.
`linkmode` **(IPoIB only)**	Sets the `linkmode` for the data-link interface. The default value is `cm`. Valid values are:

`cm` (the default)	Connected Mode. This mode uses a default MTU of `65520` bytes. and supports a maximum MTU of `65535` bytes.
`ud`	Unreliable Datagram Mode. If Connected Mode is not available for a remote node, Unreliable Datagram mode is automatically used instead. This mode uses a default MTU of `2044` and supports a maximum MTU of `4092` bytes.

`allowed-address`	Configure an IP address for the exclusive-IP zone and also limit the set of configurable IP addresses that can be used by an exclusive-IP zone. To specify multiple addresses, use a list of comma-separated IP addresses.
`defrouter`	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.

When the `zonecfg` command creates a zone using the `SYSdefault` template, an `anet` resource with the following properties is automatically included in the zone configuration if no other IP resources are set. The `linkname` is automatically created over the physical Ethernet link and set to the first available name of the form `net`*N*, `net`*0*. To change the default values, use the `zonecfg` command.

When the default policy `auto` is used, an appropriate `mac-address` is assigned:

Oracle Solaris Zone	random `mac-address`
Oracle Solaris Kernel Zone	random `mac-address`
Oracle Solaris Zone under kernel zone	factory `mac-address`
Oracle VM Server for SPARC guest domain	factory `mac-address`
Oracle Solaris Kernel Zone running on Oracle VM Server for SPARC guest domain	factory `mac-address`

The default policy creates an automatic VNIC over the physical Ethernet link, for example, `net0`, and assigns the MAC address to the VNIC. The optional `lower-link` property is set to the underlying link, `vnic1`, over which the automatic VNIC is to be created. VNIC properties such as the link name, underlying physical link, MAC address, bandwidth limit, as well as other VNIC properties, can be specified by using the `zonecfg` command. Note that `ip-type=exclusive` must also be specified.

```
zonecfg:my-zone> set ip-type=exclusive
zonecfg:my-zone> add anet
zonecfg:my-zone:anet> set linkname=net0
zonecfg:my-zone:anet> set lower-link=auto
zonecfg:my-zone:anet> set mac-address=random
zonecfg:my-zone:anet> set link-protection=mac-nospoof
zonecfg:my-zone:anet> end
```

The following example shows a `solaris` brand zone configured with an IPoIB data-link interface over the physical link `net5` with the IB partition key `0xffff`:

```
zonecfg:my-zone> set ip-type=exclusive
zonecfg:my-zone:anet> add anet
zonecfg:my-zone:anet> set linkname=ib0
zonecfg:my-zone:anet> set lower-link=net5
zonecfg:my-zone:anet> set pkey=0xffff
zonecfg:my-zone:anet> end
```

For more information on properties, see the `zonecfg`(1M) man page. For additional information on the link properties, see the `dladm`(1M) man page.

`net` — `address`, `allowed-addressphysical`, `defrouter`

Note - For a shared-IP zone, both the IP address and the physical device must be 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.

In the following example for a shared-IP zone, the physical interface `nge0` is added to the zone with an IP address of `192.168.0.1`. To list the network interfaces on the system, type:

```
global# ipadm show-if -po ifname,class,active,persistent
lo0:loopback:yes:46--
nge0:ip:yes:----
```

Each line of the output, other than the loopback lines, will have the name of a network interface. Lines that contain `loopback` in the descriptions do not apply to cards. The `46` persistent flags indicate that the interface is configured persistently in the global zone. The `yes` active value indicates that the interface is currently configured, and the class value of `ip` indicates that `nge0` is a non-loopback interface. The default route is set to `10.0.0.1` for the zone. Setting the `defrouter` property is optional. Note that `ip-type=shared` is required.

```
zonecfg:my-zone> set ip-type=shared
zonecfg:my-zone> add net
zonecfg:my-zone:net> set physical=vnic1
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 VNIC is used for the physical interface, which is a VLAN. To determine which data-links are available, use the command `dladm show-link`. The `allowed-address` property constrains the IP addresses that the 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=10.1.1.32/24
zonecfg:my-zone:net> set physical=vnic1
zonecfg:myzone:net> set defrouter=10.1.1.1
zonecfg:my-zone:net> end
```

Only the physical device type will be specified in the `add net` step. The `physical` property can be a VNIC.

Note - The Oracle™ Solaris operating system supports all Ethernet-type interfaces, and their data-links can be administered with the `dladm` command.

`device` — `match`, `allow-partition`, `allow-raw-io`

The device name to match can be a pattern to match or an absolute path. Both `allow-partition` and `allow-raw-io` can be set to `true` or `false`. The default is `false`. `allow-partition` enables partitioning. `allow-`

`raw-io` enables `uscsi`. For more information on these resources, see `zonecfg`(1M).

Restrictions on what can be specified in the `device:match` resource property for `solaris-kz` zones include the following:

- Only one resource is allowed per LUN.
- Slices and partitions are not supported.
- Support is only provided for raw disk devices.
- The supported device paths are `lofi`, `ramdisk`, `dsk`, and `zvols`.

In the following example, `uscsi` operations on a disk device are included in a `solaris` zone configuration.

```
zonecfg:my-zone> add device
zonecfg:my-zone:device> set match=/dev/*dsk/cXtYdZ*
zonecfg:my-zone:device> set allow-raw-io=true
zonecfg:my-zone:device> end
```

Veritas volume manager devices are delegated to a non-global zone by using `add device`.

In the following example, a storage device is added to a `solaris-kz` zone:

```
zonecfg:my-zone> add device
zonecfg:my-zone:device> set storage=iscsi:///
luname.naa.600144f03d70c80000004ea57da10001
zonecfg:my-zone:device> set bootpri=0
zonecfg:my-zone:device> end
```

Caution - Before adding devices, see "Device Use in Non-Global Zones" in "Creating and Using Oracle Solaris Zones ", "Running Applications in Non-Global Zones" in "Creating and Using Oracle Solaris Zones ", and "Privileges in a Non-Global Zone" in "Creating and Using Oracle Solaris Zones " 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" in "Creating and Using Oracle Solaris Zones ". 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, "About Resource Controls," in "Administering Resource Management in Oracle Solaris 11.2 " and "Resource Controls Used in Non-Global Zones" in "Creating and Using Oracle Solaris Zones ".

`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.

For more information, see the `tecla`(5) man page.

Glossary

auxiliary zone state Used to communicate additional state information to the global zone. See also zone state.

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.

CMT resources CPUS, cores, and sockets.

CPU In the zones context, refers to a hardware thread.

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 Oracle 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 The root user or an administrator with the root 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 software partitioning technology that provides a complete runtime environment for Solaris 10 applications executing in a `solaris10` branded zone on a system running the Oracle Solaris 11 release.

Oracle Solaris Kernel Zones A software partitioning technology that provides a full kernel and user environment within a zone, and also increases kernel separation between the host and the zone.

Oracle Solaris Zones A software partitioning technology used to virtualize operating system services and provide an isolated, secure environment in which to run applications.

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.

read-only zone An Immutable Zone configured with a read-only root.

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, unavailable, running, or shutting down.

Index