Oracle® Solaris Zones Configuration Resources

Part No: E57855
October 2017

ORACLE

Oracle Solaris Zones Configuration Resources

Part No: E57855

Référence: E57855

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` ou le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` si vous êtes malentendant.

Contents

Using This Documentation 7

1 Configuration Resources for Non-Global Zones 9
About Resources in Zones 9
Using Rights Profiles and Roles in Zone Administration 10
`zonecfg template` Property and Tokens 11
Zone Pre-Installation Configuration Process 13
Configurable Resources and Properties for Zones 14
Zone Name 14
Zone Path 14
Zone Autoboot 14
`solaris` and `solaris10` Only: `global-time` Property 15
`file-mac-profile` Property for Immutable Zones 15
`admin` Resource for Zones 16
`dedicated-cpu` Zone Resource 17
`solaris-kz` Only: `virtual-cpu` Resource 18
`capped-cpu` Zone Resource 19
Scheduling Class 19
`capped-memory` Resource and Physical Memory Control 20
`solaris` and `solaris10` Only: `npiv` Resource 22
`solaris` and `solaris10` Only: `rootzpool` Resource 23
Adding a `zpool` Resource Automatically 25
`solaris-kz` SPARC Only: Kernel Zone Migration Class and Host Compatibility Level 26
Zone Network Interfaces 28
File Systems Mounted in Zones 34
File System Mounts and Updating 35
Host ID in Zones 35

/dev File System in Non-Global Zones 36
Removable lofi Device in Non-Global Zones 36
Disk Format Support in Non-Global Zones 37
Kernel Zones Device Resources With Storage URIs 37
Configurable Privileges in Zones 39
Associating Resource Pools With Zones 39
Setting Zone-Wide Resource Controls 40
Including a Comment for a Zone 43
About Using the zonecfg Command 43
zonecfg Modes 44
zonecfg Command in Command-File Mode 44
zonecfg Command in Interactive Mode 44
Zone Configuration Data 47
Resource Types and Properties 47
Resource Type Properties 53
Tecla Library and Non-Global Zones 67

Index 69

Using This Documentation

- **Overview** – Reference guide for Oracle Solaris zone configuration resources and properties used with Oracle Solaris Zones.
- **Audience** – Technicians, system administrators, and authorized service providers.
- **Required knowledge** – Experience administering Oracle Solaris environments. Experience with virtualized environments is a plus.

Product Documentation Library

Documentation and resources for this product and related products are available at http://www.oracle.com/pls/topic/lookup?ctx=E53394-01.

Feedback

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

♦ ♦ ♦ CHAPTER 1

Configuration Resources for Non-Global Zones

This chapter provides an introduction to the `zonecfg` command resources and properties used in non-global zone configuration.

The following topics are covered in this chapter:

- "About Resources in Zones" on page 9
- "Using Rights Profiles and Roles in Zone Administration" on page 10
- "Zone Pre-Installation Configuration Process" on page 13
- "Configurable Resources and Properties for Zones" on page 14
- "About Using the `zonecfg` Command" on page 43
- "Zone Configuration Data" on page 47
- "Tecla Library and Non-Global Zones" on page 67

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.

In the Oracle Solaris 11.3 release, the default `solaris` branded zone is referred to as a native zone.

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 system 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

The `root` user has all administrative rights. The `root` user can assign administrative rights to users, such as a rights profile, a role, or specific privileges and authorizations.

- For information about using assigned rights in Oracle Solaris, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.3*.
- For information about zones authorizations, see "`admin` Resource for Zones" on page 16.
- For information about using privileges in a non-global zone, see "Privileges in a Non-Global Zone" in *Creating and Using Oracle Solaris Zones*.

The zones rights profiles are:

Zone Security rights profile

For administrators who will create and configure zones.

The Zone Security rights profile includes the `zonecfg` or `txzonemgr` commands and every `solaris.zone.*` authorization. The assignee can delegate zone administration. For information about `txzonemgr`, see "Creating Labeled Zones" in *Trusted Extensions Configuration and Administration*.

If the `auths` property of the `admin` resource is configured in the managed zone, this rights profile is not sufficient to create, log in, and configure zones. The zone administrator must be named in the `user` property of the `admin` resource and be assigned the `solaris.zone.*` authorizations.

Note - This rights profile permits the user to create or modify or delete any zone configuration on the host.

Zone Configuration rights profile

For administrators who will create and modify zones.

The Zone Configuration rights profile enables a zone administrator to configure a zone. For a migrated zone, the administrator must be granted this rights profile on the target system to complete the migration if a configuration for the zone does not already exist on the target system. The Zone Configuration rights profile includes the `zonecfg` command only.

If the `auths` property of the `admin` resource is configured in the managed zone, this rights profile is not sufficient to configure zones. The zone administrator must be named in the `user` property of the `admin` resource and be assigned the `solaris.zone.config` authorization. If login is restricted, the zone administrator must also be assigned the `solaris.zone.login` authorization.

Note - This rights profile permits the user to create or modify or delete any zone configuration on the host.

Zone Management rights profile

For administrators who will manage existing zones.

The Zone Management rights profile includes the `zlogin` and `zoneadm` commands.

If the `auths` property of the `admin` resource is configured in the managed zone, this rights profile is not sufficient to manage zones. The zone administrator must be named in the `user` property of the `admin` resource and be assigned the `solaris.zone.*` authorizations to log in and manage the zone.

Zone Migration rights profile

For administrators who will migrate any type of zone.

The Zone Migration rights profile enables a zone administrator to perform migration of an `installed` or `running` zone. A zone administrator who is assigned this profile can perform live or warm migrations. The Zone Migration rights profile includes the `zoneadm` and `zonecfg` commands.

If the `auths` property of the `admin` resource is configured in the managed zone, this rights profile is not sufficient to migrate zones. The zone administrator must be named as a `user` in the `admin` resource and be assigned the `solaris.zone.migrate` authorization. If login is restricted, the zone administrator must also be assigned the `solaris.zone.login` authorization.

To use the profiles, see "`admin` Resource for Zones" on page 16. Also see the `profiles(1)` and `prof_attr(4)` man pages for information about zones profiles.

For information about Oracle Solaris features that protect applications running on your system, see "Protecting and Isolating Applications" in *Oracle Solaris 11.3 Security and Hardening Guidelines*.

`zonecfg template` Property and Tokens

Use different templates to get a specific brand, to get an empty configuration, or to create a zone configuration that looks like another zone that has already been configured.

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 1 `zonecfg template` Tokens

Token	Description	Usage
%{*zonename*}	The name of the zone.	Can be used in `zonecfg` as input from the user, or input from a template value.
%{*id*}	A unique instance number that is the resource's *id* property value.	Can be used in `zonecfg` as input from the user, or input from a template value. Evaluates to the `id` property of a particular resource. Should be used within a resource scope that supports the `id` property.
%{ *global-rootzpool*}	Evaluates to the name of the root pool in the global zone.	Used in the default `solaris-kz` `device` resource.
%%	Evaluates to %.	Can be used in `zonecfg` as input from the user.

TABLE 2 Supported Tokens by Resource Property

Resource	Property	Supported Tokens
global property	`zonepath`	%{*zonename*}
`anet`	`linkname`	%{*id*}
`dataset`	`name`	%{*zonename*}
`device`	`match`	%{*zonename*}, %{*id*}, %{*global-rootzpool*}
	`storage`	%{*zonename*}, %{*id*}, %{*global-rootzpool*}
`fs`	`dir`	%{*zonename*}
`net`	`physical`	%{*id*}
`rootzpool`	`storage`	%{*zonename*}, %{*global-rootzpool*}
`suspend`	`storage`	%{*zonename*}, %{*global-rootzpool*}
	`path`	%{*zonename*}
`zpool`	`storage`	%{*zonename*}, %{*global-rootzpool*}

EXAMPLE 1 %{*zonename*} Property in `zonepath`

```
zonecfg:nz> info zonepath
zonepath.template: /system/zones/%{zonename}
zonepath: /system/zones/nz
zonecfg:nz> set zonename=new-zone
zonecfg:new-zone> info zonepath
zonepath.template: /system/zones/%{zonename}
zonepath: /system/zones/new-zone
```

```
zonecfg:new-zone>
```

EXAMPLE 2 Token Used for the `storage` Property in the `solaris-kz` `device` Resource

```
device 0:
         match not specified
         storage.template: dev:/dev/zvol/dsk/%{global-rootzpool}/VARSHARE/zones/%
{zonename}/disk%{id}
         storage: dev:/dev/zvol/dsk/rpool/VARSHARE/zones/kernel-zone1/disk0
         id: 0
         bootpri: 0
```

Note - You can configure a `solaris-kz` branded zone by using the `SYSsolaris-kz` template. By default, the `SYSsolaris-kz` template configures a zone with 4 virtual CPUs and 4 gigabytes (GB) of memory.

You can configure a minimal kernel zone by using the `SYSsolaris-kz-minimal` template. The `SYSsolaris-kz` minimal template configures a zone with 1 virtual CPU and 2 GB of memory.

The 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.

Zone Pre-Installation Configuration Process

Before you can install a non-global zone and use it on your system, the zone must be configured.

The `zonecfg` command is used to create the configuration and to determine whether the specified resources and properties are valid on a hypothetical system. The check performed by `zonecfg` for a given configuration verifies the following:

- For `solaris` and `solaris10` branded zones, 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.

Configurable Resources and Properties for Zones

This section covers the required and optional zone resources and properties that can be configured. Only the zone name is required. Additional information is provided in "Zone Configuration Data" on page 47. For more information about configuration options that are specific to a particular brand of zone, see the `solaris(5)` and `solaris-kz(5)` man pages.

Zone Name

You must choose a name for your zone.

Zone Path

If you do not specify the path, the default value of `zonepath` is `/system/zones/%{`*zonename*`}`. If the zone configuration does not have a `rootzpool` resource, the ZFS dataset `%{`*global-rootzpool*`}/VARSHARE/system/zones/%{`*zonename*`}`, is created and mounted at `/system/zones/%{`*zonename*`}`.

If you choose 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. The parent of the `zonepath` must be a ZFS dataset only if the `zonepath` dataset is not automatically created.

Kernel zones do not support the `zonepath` property. The zone root is contained within a ZFS volume. The device onto which the zone is installed is specified with a `device` resource that has the `bootpri` property set to any positive integer value.

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.

`solaris` and `solaris10` Only: `global-time` Property

Set the `global-time` property to specify whether you want to allow changing either the zone-specific time or the system-wide time from within the non-global zone.

- A value of `global-time=true` for the `global-time` property indicates that the zone is allowed to set system-wide time.
- A value of `global-time=false` for the `global-time` property indicates the zone is allowed to set zone-specific time.

EXAMPLE 3 Enabling Zone to Set Zone-Specific Time

```
# zonecfg -z my-zone
zonecfg:my-zone> set global-time=false
zonecfg:my-zone> exit
```

You should assign a value for the `global-time` property. However, if the value is not set but the `sys_time` privilege is explicitly assigned using the `limitpriv` property, the value of the `global-time` is treated as `true`. If the `sys_time` privilege is not explicitly assigned by using the `limitpriv` property, `global-time` is treated as `false`.

Depending on the `global-time` property setting in Oracle Solaris 11.3, a non-global zone process with the `sys_time` privilege can manipulate either the virtual zone-specific time or the system-wide time by using the following system calls.

- `stime(2)`
- `clock_settime(3C)`
- An IA-specific real-time clock (RTC) call to write time of day clock

See "Privileges in a Non-Global Zone" in *Creating and Using Oracle Solaris Zones* for more information on privileges.

`file-mac-profile` Property for Immutable Zones

Use the `file-mac-profile` to configure Immutable Zones with read-only roots.

For more information, see Chapter 11, "Configuring and Administering Immutable Zones" in *Creating and Using Oracle Solaris Zones*.

`admin` Resource for Zones

The `admin` setting allows you to set zone administration authorization. The preferred method for defining authorizations is through the `zonecfg`command.

`auths`

Specify the authorizations for the user name.

The values for `auths` are:

`solaris.zone.clonefrom`

If RBAC is in use, allows the specified zone to be used as a source from which to clone a new zone. Subcommands that make a copy of another zone require the authorization `solaris.zone.clonefrom/`*source_zone*.

`solaris.zone.config`

If RBAC is in use, allows modification of the persistent configuration of the zone by using the authorization `solaris.zone.config/`*zonename*. For more information on the persistent configuration, see Chapter 6, “Live Zone Reconfiguration” in *Creating and Using Oracle Solaris Zones*.

`solaris.zone.liveconfig`

If RBAC is in use, allows inspection and modification of the live zone configuration by using the authorization `solaris.zone.liveconfig/`*zonename*. For more information on the live zone configuration, see Chapter 6, “Live Zone Reconfiguration” in *Creating and Using Oracle Solaris Zones*.

`solaris.zone.login`

If RBAC is in use, allows authenticated use of `zlogin` into this zone. The authorization `solaris.zone.login/zonename` is required for interactive logins. Password authentication takes place in the zone. For more information, see `zlogin`(1) and Chapter 4, “About Non-Global Zone Login” in *Creating and Using Oracle Solaris Zones*.

`solaris.zone.manage`

If RBAC is in use, allows normal management of the configured zone. For non-interactive logins, or to bypass password authentication, the authorization `solaris.zone.manage/zonename` is required.

`user`

Specify the user name.

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

dedicated-cpu Zone Resource

Use the dedicated-cpu resource to specify 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 dynamically creates 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.

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, the importance default is 1. For more information, see "pool.importance Property Constraint" in *Administering Resource Management in Oracle Solaris 11.3*.

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 11.3*.

Use the following properties 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.

To eliminate inconsistent results across system reboots, use `dedicated-cpu:cpus` to specify the exact CPUs to use. Use the `dedicated-cpu` resource instead of the automatic `virtual-cpu` resource, which only specifies `ncpus`.

Note - The `capped-cpu` resource and the `dedicated-cpu` resource are incompatible. The `cpu-shares` resource control 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 not 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 virtual CPUs (VCPUs) if you want to assign a number other than the default.

The default kernel zone configuration has 4 VCPUs. Each `virtual-cpu` can use up to 1 CPU of compute power, but could get less if there is contention for system CPU resources. The CPUs allocated to the kernel zone are defined by the `ncpus` value. You can add more CPUs to the kernel zone by adding the `virtual-cpu` property.

If a kernel zone is in a pool that was created by using the `dedicated-cpu` or the `pool` resource, then the number of virtual CPUs created match the size of that pool. Note that VCPUs are not sized based on the number of FSS shares.

If CPU resources are shared between a number of consumers, there might be periods of time when the system "de-schedules" all or part of the kernel zone.

Stolen time indicates the time when the kernel zone cannot run because the system might be using CPU resources for other purposes.

The CPU accounting state `CMS_STOLEN` displays the time a CPU spends in this state. The time is always zero for systems running on physical hardware. For CPUs running as part of a kernel zone, a non-zero value of this state reflects the amount of time a virtual CPU did not actually have access to a physical CPU. Stolen time is reported by `zonestat(1)`, `mpstat(1M)`, `iostat(1M)`, `vmstat(1M)`, and other utilities.

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. You do not need to set both the `dedicated-cpu` and the `virtual-cpu` resources.

`capped-cpu` Zone 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` Zone Resource” on page 17.

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) is 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 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 11.3* and "How to Associate a Pool With a Scheduling Class" in *Administering Resource Management in Oracle Solaris 11.3* .
- If the `cpu-shares` resource control 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` command described in the `priocntl(1)` man page to move running processes into a different scheduling class without changing the default scheduling class and rebooting.

`capped-memory` Resource and Physical Memory Control

To use the `capped-memory` resource, the `resource-cap` package must be installed in the global zone. Also see capped-memory in "Resource Types and Properties" on page 47.

`solaris` Zones and the `capped-memory` Zone Resource

For native (`solaris`) branded zones, the `capped-memory` resource sets limits for `physical`, `swap`, and `locked` memory properties. Each limit is optional, but at least one must be set.

- Determine values for the `physical` property if you plan to cap memory for a native 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 `physical` property of the `capped-memory` resource represents a soft RAM allocation limit that is enforced by `rcapd`. If a zone hits its physical limit, the zone can continue to allocate RAM, but paging to the swap device will occur even when there's no overall memory shortfall on the system. Paging can generate large amounts of I/O, which can

negatively impact other operations on the system. In contrast, limiting swap has no direct impact on the paging activity of the system. Setting `swap` without setting `physical` can be an effective way to limit the amount of memory used by a native zone.

- When you limit the amount of swap a zone can allocate, you also limit the amount of RAM the zone can allocate. A zone cannot allocate more RAM than it has swap. If a zone hits its swap limit, new memory allocations in that zone will fail even when there is no overall memory shortfall on the system.

  The `swap` property of the `capped-memory` resource is the preferred way to set the `zone.max-swap` resource control for a native zone.

- The `locked` property of the `capped-memory` resource is the preferred way to set the `zone.max-locked-memory` resource control for a native zone.

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 the following documentation:

- Chapter 10, "About Controlling Physical Memory by Using Resource Capping" in *Administering Resource Management in Oracle Solaris 11.3*
- Chapter 11, "Administering the Resource Capping Daemon" in *Administering Resource Management in Oracle Solaris 11.3*
- "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 11.3*.

`solaris-kz` Zones and the `capped-memory` Resource

For kernel zones, the `physical` property is required. The `physical` property represents the amount of RAM reserved for the kernel zone's memory. When you specify the `physical` property, you can also specify the `pagesize-policy` property, which sets the policy for using large pages for physical memory.

For kernel zones, the `swap` and `locked` limits are not allowed. The `rcapd` utility is not used. Live Zone Reconfiguration is not supported.

The default `SYSsolaris-kz` template sets the `pagesize-policy` to `largest-available`, which is the recommended value.

Oracle Solaris systems that do not support the `pagesize-policy` property use a compatible default pagesize. Clearing the `pagesize-policy` property is required to live migrate a kernel zone to an older Oracle Solaris instance, or to resume a kernel zone on an older Oracle Solaris instance.

The `pagesize-policy` property values are described in "Resource Type Properties" on page 53.

See "Managing Kernel Zone Memory" in *Creating and Using Oracle Solaris Kernel Zones* for more information about these properties and examples for setting them.

For more information, also see the `solaris-kz(5)` man page.

`solaris` and `solaris10` Only: `npiv` Resource

The `npiv` resource supports N_Port_ID Virtualization (NPIV) in Oracle Solaris Zones and Oracle Solaris 10 Zones. The `npiv` resource is used to configure zones that have fibre channel devices as back-end storage for the zone root file system, and use other devices for data.

The following example delegates two `npiv` resources to the zone *my-zone*. Both `virtual-port-wwn` and `over-hba` are optional. The two `npiv` ports are automatically created during zone installation.

```
zonecfg:my-zone> add npiv
zonecfg:my-zone:npiv> set virtual-port-wwn=2100000000000001
zonecfg:my-zone:npiv> set over-hba=c9
zonecfg:my-zone:npiv> end
zonecfg:my-zone> add npiv
zonecfg:my-zone:npiv> end
zonecfg:my-zone>
```

Disks visible through the NPIV port are also visible inside the zone. Disks added to the fabric are visible automatically from within the zone. Disks removed from the fabric are automatically removed from the zone view.

The `virtual-port-wwn` property type is optional for the `npiv` resource type. It contains the port world wide name (PWWN) for the `npiv` port to be created. The port is automatically generated if not specified by users. To override the default `virtual-port-wwn` property value, use the following command from inside the `npiv` resource scope:

```
zonecfg:my-zone:npiv> set virtual-port-wwn=World Wide Name
```

The `zonecfg` command verifies that the string is valid.

`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 13, "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*.

`solaris-kz` SPARC Only: Kernel Zone Migration Class and Host Compatibility Level

Only features enabled by both migration class and host compatibility level are visible to a kernel zone. To migrate a kernel zone, you must ensure that the feature set visible to the kernel zone matches on both the source and target hosts by configuring the migration class `cpu-arch` and the `host-compatible` properties.

If not set, the default value of `cpu-arch` is *native*. The zone boots with the same CPU class as the host. You can migrate the zone between CPU types that are compatible with the CPU class of the host. By default, Silicon Secured Memory (SSM), also known as ADI, is turned off for a kernel zone.

`solaris-kz` SPARC Only: Cross-CPU Migration

Use the `cpu-arch` global property to configure kernel zones with a specific CPU class. The CPU class can be independent of the host CPU class, to ensure a safe migration between different CPU types. If an Oracle VM Server for SPARC guest domain is booted with a specific class, the guest can be migrated safely among all platforms with compatible CPU types. Kernel zones use the same set of CPU classes as guest domains.

If not set, the default value of `cpu-arch` is *native*. The zone boots with the same CPU class as the host. You can migrate the zone between CPU types that are compatible with the CPU class of the host.

The host does not resume a zone previously suspended on an incompatible platform. The host also does not boot a zone if the migration class is set to an incompatible value for the host platform. For example, a guest on a T5 will not boot if `cpu-arch` is set to `sparc64-class1`. The CPU class of the zone cannot exceed the limits of the CPU class of the host.

A kernel zone booted with the `generic` class cannot be migrated to systems earlier than the SPARC T4. Kernel zones run on SPARC T4 and Fujitsu SPARC M12, Fujitsu M10, or SPARC M10, and later supported systems.

```
cpu-arch={generic | migration-class1 | sparc64-class1}
```

The values are:

`generic`

Kernel zone can perform a CPU-type-independent migration between systems newer than T4.

`migration-class1`

Kernel zone can perform cross-CPU type migration between SPARC T4, SPARC T5, SPARC T7, SPARC S7, SPARC M5, SPARC M6, and SPARC M7.

`sparc64-class1`

Kernel zone can perform cross-CPU type migration between Fujitsu SPARC M12, Fujitsu M10, and SPARC M10.

Setting and checking the `cpu-arch` property:

```
$ zonecfg -z vzl
zonecfg:vzl> info cpu-arch
cpu-arch: generic
zonecfg:vzl> set cpu-arch=migration-class1
zonecfg:vzl> info cpu-arch
cpu-arch: migration-class1
zonecfg:vzl> exit
```

`solaris-kz` SPARC Only: `host-compatible` Property

Use the `host-compatible` property `adi` to enable the Silicon Secured Memory (SSM) feature, also known as ADI. By default, SSM is turned off for a kernel zone. To enable SSM, you must set the `host-compatible` modifier. In the global zone, on SSM capable hardware, SSM is always turned on.

If no value is set, the default host compatibility level of a kernel zone includes only features supported in the Oracle Solaris 11.2 release.

The host-compatibility levels are as follows:

- `adi` – Set the `adi` modifier to enable the SSM feature. The `adi` modifier can only be used with the default compatibility level.

  ```
  host-compatible=adi
  ```

  The `host-compatible` modifier cannot be used to enable SSM if the SSM feature is not supported by the migration class.
- `level1` – If all of your systems are running the Oracle Solaris 11.3 release, set the `host-compatible` property to `level1`, which allows enabling of all features available in the release. The `level1` level includes SSM, SPARC M7 DAX Data Analytics Accelerator (DAX) coprocessors, and VA Mask features. On DAX capable hardware, DAX is always turned on in the global zone. The `level1` setting might prevent the kernel zone from being migrated to other hosts that are running an older release of Oracle Solaris.

```
host-compatible=level1
```

- `native` – Set the `native` host compatibility level to support all features in the current version of Oracle Solaris, including SSM. Note that the `native` host compatibility level might prevent the kernel zone from being migrated to a host running a different release of Oracle Solaris.

  ```
  host-compatible=native
  ```

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

If some addresses must be automatically configured and other addresses must be available to be brought online and offline within the zone, multiple `anet` resources can be used. For example, the following configuration has two `anet` resources. The first automatically configures the `192.0.2.3` on one of the zone's interfaces. The second allows the zone to configure only `192.0.2.100` and `192.0.2.101` on the other interface.

```
zonecfg:my-zone> select anet linkname=net0
zonecfg:my-zone:anet> set allowed-address=192.0.2.3/24
zonecfg:my-zone:anet> set configure-allowed-address=true
zonecfg:my-zone:anet> end
zonecfg:my-zone> add anet
zonecfg:my-zone:anet> set allowed-address=192.0.2.100/24,192.0.2.101/24
```

```
zonecfg:my-zone:anet> set configure-allowed-address=false
zonecfg:my-zone:anet> end
zonecfg:my-zone>
```

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 11.3* 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) in zones, see the `anet` description in "Resource Type Properties" on page 53.

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:

- `allowed-address`
- `defrouter`
- `lower-link`
- `mac-address`
- `maxbw`
- `mtu`

- `priority`
- `vlan-id`

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

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

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 11.3*.

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.

  Note that an assigned data-link enables the `snoop` command to be used.

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`.

If some addresses must be automatically configured and some addresses must be able to be brought online and offline within the zone, multiple `anet` resources can be used. For example, this configuration will have two `anet` resources. The first `anet` resource automatically configures the address `192.168.3.3` on one of the zone's interface. The second `anet` resource permits the zone to configure only `192.168.3.100` and `192.168.3.101` on the other interface.

```
zonecfg:my-zone> select anet linkname=net0
zonecfg:my-zone:anet> set allowed-address=192.168.3.3/24
zonecfg:my-zone:anet> set configure-allowed-address=true
zonecfg:my-zone:anet> end
zonecfg:my-zone> add anet
zonecfg:my-zone:anet> set allowed-address=192.168.3.100/24,192.168.3.101/24
zonecfg:my-zone:anet> set configure-allowed-address=false
zonecfg:my-zone:anet> end
zonecfg:my-zone>
```

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 *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 4 How to Enable the `rds` Service in a Non-Global Zone

1. To enable RDSv3 service in an exclusive-IP or shared-IP zone, log in to the zone with the `zlogin` command 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 `root` user 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 system.

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.

Do *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 native 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 permitted 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 11.3 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 physical 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 47 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 /dev file system for the zone. 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 read-only 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 the zonecfg command 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 5 Create a Removable lofi Device With an Associated File

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

EXAMPLE 6 Create an Empty Removable lofi Device

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

EXAMPLE 7 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 3, “Zone-Wide Resource Controls,” on page 41.

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 53 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:

- Devices that are used as disks are supported. This support includes whole physical disks, whole physical or virtual disks on a SAN, devices in conjunction with Oracle Solaris Cluster, and ZFS volumes.
- Kernel zones also support NFS-based storage objects through nfs: URI.

  The NFS URI specifies an object based on a lofi device created on the given NFS file. The NFS file is accessed with credentials derived from *user* and *group*. user and group can be given as user names or as user IDs. The host can be given as an IPv4 address, as an IPv6 address, or as a host name. IPv6 addresses must be enclosed in square brackets ([]).

  Format:

  nfs://*user*:*group*@*host*[:*port*]/*nfs-share-path*/*file*

  Examples:

  ```
  nfs://admin:staff@host/export/test/nfs_file
  nfs://admin:staff@host:1000/export/test/nfs_file
  ```
- Kernel zones support 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.

  Only set `bootpri` on devices that must be bootable.
- `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.

EXAMPLE 8 Configuring a Storage URI to Create a Portable Zone Configuration

A device resource can also be used to configure a storage URI that makes the zone configuration portable to other systems.

```
# zonecfg -z my-zone
zonecfg:my-zone> add device
zonecfg:my-zone:device> set storage=nfs://user1:staff@host1/export/file1
zonecfg:my-zone:device> set create-size=4g
```

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

EXAMPLE 9 Viewing the Current Device Resources Configuration

To view information about the current configuration for device resources, use the `info` subcommand. For example:

```
$ zonecfg -z my-zone info device
device:
    match not specified
    storage: dev:/dev/zvol/dsk/rpool/VARSHARE/zones/my-zone/disk0
    id: 0
    bootpri: 0
device:
    match not specified
    storage: nfs://user1:staff@host1/export/file1
    create-size: 4g
```

You can display the output for a specific zone by specifying the ID for the zone:

```
$ zonecfg -z my-zone info device id=1
device:
    match not specified
    storage: nfs://user1:staff@host1/export/file1
```

```
create-size: 4g
id: 1
bootpri not specified
```

Configurable Privileges in Zones

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).

Associating Resource Pools With Zones

If you have configured resource pools on your system as described in Chapter 13, "Creating and Administering Resource Pools" in *Administering Resource Management in Oracle Solaris 11.3*, 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 11.3* 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 11.3* 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 11.3*. 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`.

Global scope. The `zone.max-adi-metadata-memory` resource controls the maximum amount of metadata allocated for Silicon Secured Memory (SSM) enabled pageable memory. SSM is also known as ADI.

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 3 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	

Control Name	Global Property Name	Description	Default Unit	Value Used For
			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)	
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-adi-metadata-memory		Total amount of memory for storing Silicon Secured Memory (SSM) metadata of pages that might be written to backing store, expressed as a number of bytes. SSM is also known as ADI.	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*.

About 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 Command in Command-File Mode

In command-file mode, input is taken from a file. The export subcommand described in "zonecfg Command in Interactive Mode" on page 44 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.

zonecfg Command in 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
```

`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.

`cancel`

Applicable only in the resource scope. End the resource specification and reset the scope to global. Any partially specified resources are not retained.

`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.

`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.

`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.

`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.

`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.

`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.

`export`

Print the configuration to standard output, or to the output file specified, in a form that can be used in a command file.

`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.

`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.

`revert`

Revert configuration back to the last committed state.

`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.

`set`

Set a given property name to the given property value. Note that some properties, such as `zonepath` used in native and `solaris10` branded zones, are global, while others are resource specific. Thus, this command is applicable in both the global and resource scopes.

`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.

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.

Resource Types and Properties

The resource and property types are described as follows:

`anet`

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

`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.

`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.

`autoshutdown`

Global scope. The action to take for this zone upon clean shutdown of the global zone. The value can be `shutdown` (a clean zone shutdown; the default); `halt`, or `suspend`.

`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*.

`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` Zone Resource" on page 19.

`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.

`solaris` and `solaris10` **Only:** `dataset`

The only dataset type that should be used with a `dataset` resource is a ZFS™ file system. Add a ZFS `dataset` resource to enable the delegation of storage administration to a non-global zone. The zone administrator can create and destroy file systems within that dataset, and modify properties of the dataset. The zone administrator can create child file systems and clones of its descendants. 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 12, "Troubleshooting Miscellaneous Oracle Solaris Zones Problems" in *Creating and Using Oracle Solaris Zones* for information on dataset issues.

Note - Use the `device` resource instead of the `dataset` resource in kernel 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` Zone Resource" on page 17.

`device`

The `zonecfg device` 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.

`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.

`solaris` and `solaris10` Only:`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 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*.

`solaris-kz` **Only:** `ib-vhca`

The `ib-vhca` resource automatically creates a temporary virtual InfiniBand HCA device for an exclusive-IP zone when the zone boots. The device is deleted when the zone halts.

Also see *Managing Network Virtualization and Network Resources in Oracle Solaris 11.3*

`ip-type`

This property is required to be set for all non-global zones. See "Exclusive-IP Non-Global Zones" on page 31, "Shared-IP Non-Global Zones" on page 30, and "How to Configure the Zone" in *Creating and Using Oracle Solaris Zones*.

`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"
```

The following entry allows you to examine and modify the resource controls associated with an active process, task, or project on the system by using the `priocntl` command:

```
global# zonecfg -z userzone
zonecfg:userzone> set limitpriv="default,proc_priocntl"
```

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.

`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.

`npiv`

Provide N_Port_ID Virtualization (NPIV) support in Oracle Solaris Zones.

`pool`

This resource 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` Zone Resource” on page 17.

`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 40 for more information.

Note - To configure zone-wide controls using the `set` *global_property_name* subcommand of `zonecfg` instead of the `rctl` resource, see “How to Configure the Zone” in *Creating and Using Oracle Solaris Zones*.

`scheduling-class`

This property sets the scheduling class for the zone. See “Scheduling Class” on page 19 for additional information and tips.

`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 18.

`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, underscores (_), 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`

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

If specified, the `zonepath` property provides 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 is 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
`/system/zones/my-zone`	`zonecfg zonepath`
`/system/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.

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:

- `clone` (`solaris.zone.clonefrom`)
- `config` (`solaris.zone.config`)
- `config` (`solaris.zone.liveconfig`)
- `login` (`solaris.zone.login`)
- `manage` (`solaris.zone.manage`)

Note that these `auths` do not allow you to create a zone. This capability is included in the Zone Security profile.

`anet`

`linkname`, `lower-link`, `allowed-address`, `allowed-mac-address`, `allowed-vlan-ids`, `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`

For information about additional `anet` properties, see the `zonecfg(1M)` man page.

`solaris-kz` **Only:** In addition to static configuration of `anet` MAC addresses and VLAN IDs, there is dynamic MAC address and VLAN ID configuration. A zone can push the MAC address and VLAN ID it requires to the host, and VNIC creation succeeds in this address.

Note - Dynamic configuration cannot be used on single root I/O-based `anet` configurations, which have the `iov` property set to `on`.

To determine which MAC prefixes and VLAN IDs are allowed, use the `dladm show-phys` command with the `-o` option:

```
# dladm show-phys -o link,media,device,allowed-addresses,allowed-vids
LINK    MEDIA       DEVICE   ALLOWED-ADDRESSES   ALLOWED-VIDS
net0    Ethernet    zvnet0   fa:16:3f,           100-199,
                             fa:80:20:21:22      400-498,500
```

- The `anet mac allowed-mac-address` property provides a set of MAC address prefixes. A kernel zone can create a VNIC with a MAC address that is one of the MAC address prefixes in the `allowed-mac-address` list. These prefixes can be 1 to 5 octets in length.

  ```
  zonecfg:kz1> add anet
  zonecfg:kz1:anet> add mac
  zonecfg:kz1:anet:mac> add allowed-mac-address fa:16:3f
  zonecfg:kz1:anet:mac> add allowed-mac-address fa:80:20:21:22
  zonecfg:kz1:anet:mac> end
  zonecfg:kz1:anet> end
  ```

  The `allowed-mac-address` property does not affect the `mac-address` property. The `allowed-mac-address` property controls the additional MAC addresses for the `anet` resource.

  You can also use the special keyword `any` to match any MAC address.

- The `anet vlan allowed-vlan-ids` property specifies the range of VLAN IDs that can be dynamically configured for that `anet`. Setting `allowed-vlan-ids` to the special keyword `any` allows the zone to use any valid VLAN ID.

  ```
  zonecfg:kz1> add anet
  zonecfg:kz1:anet> add vlan
  zonecfg:kz1:anet:vlan> add allowed-vlan-ids 100-199
  zonecfg:kz1:anet:vlan> add allowed-vlan-ids 400-498
  zonecfg:kz1:anet:vlan> add allowed-vlan-ids 500
  zonecfg:kz1:anet:vlan> end
  ```

```
zonecfg:kz1:anet> end
```

The `allowed-vlan-ids` property does not affect the `anet` `vlan-id` property. The `allowed-vlan-ids` property only controls the additional VLAN IDs for the `anet` resource.

`solaris-kz` **Only:** You can create and administer single root I/O (SR-IOV) NIC virtual functions (VF) on kernel zones by using the `zonecfg` `anet` resource `iov` property. Do not set the `iov` property to `auto` or `on` if any of the following properties are set:

- `allowed-address`
- `allowed-dhcp-cids`
- `configure-allowed-address`
- `cos`
- `defrouter`
- `etsbw-lcl`
- `evs`
- `link-protection`
- `maxbw`
- `mtu`
- `priority`
- `rxfanout`
- `rxrings`
- `txrings`
- `vlan-id`
- `vport`
- `vsi-mgrid`
- `vsi-typeid`
- `vsi-vers`

If the `iov` property is already set to `auto` or `on`, then setting any of these properties fails.

For examples and more information, see "Managing Single-Root I/O NIC Virtualization on Kernel Zones" in *Creating and Using Oracle Solaris Kernel Zones* and the `zonecfg`(1M) man page.

Note - For kernel zone warm migrations, suspend and resume operations are not supported if the `zonecfg` `iov` property is set to `auto` or `on`. For further information on kernel zone suspend and resume operations, see "Configuring the suspend Resource" in *Creating and Using Oracle Solaris Kernel Zones* and "Using Warm Migration to Migrate a Kernel Zone" in *Creating and Using Oracle Solaris Kernel Zones*.

`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.

`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.

`iov`

See "Managing Single-Root I/O NIC Virtualization on Kernel Zones" in *Creating and Using Oracle Solaris Kernel Zones*. For specific information on shadow VNICS used to provide network statistics, see "Using Virtual Functions and Shadow VNICs With Oracle Solaris Kernel Zones" in *Creating and Using Oracle Solaris Kernel Zones*.

`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.

`linkname`

Specify a name for the automatically created VNIC interface or IPoIB interface. Note that `solaris-kz` does not support IPoIB.

`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.

`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.

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
```

The following example shows how to configure VLANs with zones. The `vlan-id` property is not supported on IPoIB datalinks.

```
zonecfg:my-zone:anet> add anet
zonecfg:my-zone:anet> set linkname=net0
```

```
zonecfg:my-zone:anet> set lower-link=net0
zonecfg:my-zone:anet> set vlan-id=101
zonecfg:my-zone:anet> end
```

For more information about properties, see the zonecfg(1M) man page. For additional information on the link properties, see the dladm(1M) man page. For information about creating and administering single root I/O (SR-IOV) NIC virtual functions (VF) on kernel zones by using the `zonecfg iov anet` property, see "Managing Single-Root I/O NIC Virtualization on Kernel Zones" in *Creating and Using Oracle Solaris Kernel 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
```

`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`, `pagesize-policy`

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
```

The `capped-memory:pagesize-policy` property values can be one of the following:

`largest-only`

Only the largest possible page size for the Kernel Zone's physical memory is allocated. If you fail to assign all the pages, then you fail to boot the zone.

`largest-available`

The largest possible page size is used, scaling down the page size if the system cannot allocate all physical memory with a particular page size. This value is the default because scaling to a usable page size ensures the zone can boot.

`smallest-only`

Lowest allowable page size required to boot the Kernel Zone for the particular platform is chosen.

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

`dataset`

`name`, `alias`

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
```

The `%{`*zonename*`}` token can be used for the *name* property.

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

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

`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
```

`device`

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

The device name to match can be a pattern to match or an absolute path. The following tokens are supported for the `match` and `storage` properties:

- `%{`*zonename*`}`
- `%{`*id*`}`
- `%{` *global-rootzpool*`}`

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 added to 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
```

If using a token for the storage property, when a new instance of the `device` resource is added to a zone configuration, the system displays:

```
    device 0:
        match not specified
        storage.template: dev:/dev/zvol/dsk/%{global-rootzpool}/VARSHARE/zones/%
{zonename}/disk%{id}
        storage: dev:/dev/zvol/dsk/rpool/VARSHARE/zones/kernel-zone1/disk0
        id: 0
        bootpri: 0
```

Because storage is the only property that has a default value, only this property contains a value in the `info` output displayed after adding the resource.

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.

`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.

`solaris-kz` **Only:** `ib-vhca`

`over-hca`, `id`, `port`

The `ib-vhca` resource specifies the physical function (PF) that is used to allocate a virtual function (VF).

Use the following steps to allocate a VF in a kernel zone:

1. Virtualize the PF by using the `ibadm` command described in the `ibadm`(1M) man page.
2. Use the `zonecfg` command to allocate a VF to a kernel zone. Note that a specific VF index is not specified. At boot time, an available VF is dynamically allocated from the specified PF to the kernel zone by `zoneadmd`. If a VF is not available, the resource allocation fails.

`id`

Unique identifier for the `ib-vhca` resource.

`over-hca`

Sets the physical InfiniBand device to use for configuration of the virtual InfiniBand device. To obtain the device name, see the `ibadm` command.

`port`

Use the port resource to specify the allowable `pkey` values for the allocated VF. The port also has an `id` property that corresponds to the physical port number, which is typically `1` or `2`.

`id`

The `id` value is used to uniquely identify the `port` resource. The `id` corresponds to the physical port number.

`pkey`

Specifies the InfiniBand Partition key value. The `pkey` value can either be a keyword or a comma-separated list of hexadecimal values. Do not use the `0x` prefix to specify the hexadecimal value.

The keyword used for `pkey` is `auto`. Use the `auto`keyword to automatically generate and assign a `pkey` value based on the `over-hca` value specified.

`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:my-zone:net> set allowed-address=10.1.1.32/24
zonecfg:my-zone:net> set physical=vnic1
zonecfg:my-zone: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. You can administer the data-links with the `dladm` command.

`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 11.3* and "Resource Controls Used in Non-Global Zones" in *Creating and Using Oracle Solaris Zones*.

`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 23. 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.

`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
```

`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 23. 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.

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 Library and Non-Global Zones

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.

Index

A

ADI
  enabling, 27
`allowed-addresses`
  exclusive-IP zone, 31
`autoboot`, 14
`autoshutdown` property, 48

B

`bootargs` property, 48

C

`capped-cpu` resource, 19, 48
`capped-memory` resource, 20, 48
configurable privileges, zone, 39
configuring with zone, 58
cross-CPU migration
  kernel zones, 26, 26

D

data-link, 29
DAX
  kernel zones, 27
`dedicated-cpu` resource, 17, 49
`defrouter`, 64
  exclusive-IP zone, 31
Device Resources
  with storage URIs, 37
DHCP
  exclusive-IP zone, 31
disk format support
  zones, 37
`dtrace_proc`, 50
`dtrace_user`, 50

E

enabling
  ADI, 27
  Silicon Secured Memory (SSM), 27
EVS
  with zones, 29
exclusive-IP zone, 31

F

fair share scheduler (FSS), 19
features
  exclusive-IP zone, 31

G

`global-time`, 15

H

host-compatibility levels
  descriptions, 27

host-compatible
  kernel zones, 26, 27
`host-compatible` property
  kernel zones, 27
`hostid`, 35

I

IP Filter
  exclusive-IP zone, 31
IP routing
  exclusive-IP zone, 31
`ip-type` property, 50
IPMP
  exclusive-IP zone, 31
IPoIB, 58

K

kernel zones
  cross-CPU migration, 26, 26
  host-compatible, 26, 27
  `host-compatible` property, 27
  `virtual-cpu` resource, 18

L

`limitpriv` property, 50
`linkmode`, 56
locked memory cap, 21
`lofi` device
  removable, 36

N

net resource
  exclusive-IP zone, 31
  shared-IP zone, 30
`npiv` property, 51
`npiv` resource
  `solaris` brand, 22

P

`pagesize-policy` property, 21
physical memory cap, 20
`pkey`, 57, 58
`pool` property, 51
properties
  `host-compatible`, 27

R

read-only zone
  `file-mac-profile`, 15
read-only zone root, 15
Reliable Datagram Sockets (RDS), 33
removable `lofi` device, 36
resource controls
  zone-wide, 40
`rootzpool` resource
  `solaris` brand, 23

S

`scheduling-class` property, 51
shared-IP zone, 30
Silicon Secured Memory (SSM)
  enabling, 27
SSM *See* Silicon Secured Memory (SSM)
swap space cap, 21
`sys_time`, 50

T

temporary pool, 17

V

`virtual-cpu` resource, 52

kernel zones, 18
VLAN, 58

Z

ZFS
dataset, 48
zone, 53
anet, 47, 53
autoshutdown property, 48
bootargs property, 48
capped-cpu, 48
capped-memory, 48
configurable privileges, 39
configuration overview, 13
configuring, 43
dataset, 48
dedicated-cpu, 49
disk format support, 37
exclusive-IP, 31
ib-vhca, 50, 63
ip-type, 50
limitpriv, 50
net, 51
npiv, 51
pool, 51
property types, 47
resource controls, 40
resource type properties, 53
resource types, 47
rights, roles, profiles, 10
rootzpool , 66
scheduling-class, 51
shared-IP, 30
virtual-cpu, 52
zone-wide resource controls, 47
zone admin authorization, 16
zone-wide resource controls, 40
zone.cpu-cap resource control, 40
zone.cpu-shares resource control, 40
zone.max-adi-metadata-memory, 41
zone.max-locked-memory resource control, 40
zone.max-lofi resource control, 40
zone.max-lwps resource control, 41
zone.max-msg-ids resource control, 41
zone.max-processes resource control, 41
zone.max-sem-ids resource control, 41
zone.max-shm-ids resource control, 41
zone.max-shm-memory resource control, 41
zone.max-swap resource control, 41
zonecfg
admin authorization, 16
autoboot, 14
entities, 47
global-time, 15
in global zone, 43
modes, 44
operations, 13
scope, 44
scope, global, 44
scope, resource specific, 44
subcommands, 44
template, 11
temporary pool, 17
zones
capped-memory resource, 20
fair share scheduler (FSS), 19
tokens, 11
zones capped-cpu, 19
zpool resource, 25