

Oracle® Solaris Zones Configuration Resources

ORACLE®

Part No: E61040
August 2021

Part No: E61040

Copyright © 2004, 2021, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

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

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle recognizes the influence of ethnic and cultural values and is working to remove language from our products and documentation that might be considered insensitive. While doing so, we are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is an ongoing, long-term process.

Référence: E61040

Copyright © 2004, 2021, Oracle et/ou ses affiliés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Inside sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Epyc, et le logo AMD sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité de la documentation, visitez le site Web Oracle Accessibility Program, à l'adresse : <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

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	9
1 Non-Global Zone Configuration Command and Resources	11
zonecfg Command	11
zonecfg and the Global Zone	12
zonecfg Templates	13
zonecfg Command-Line Editing	15
About Resources in Zones	15
Configurable Resource Types and Global Properties	15
Scheduling Class	16
Capped Memory and Physical Memory Control	16
Kernel Zone Migration Class and Host Compatibility Level (solaris-kz Only)	18
About Zone Network Interfaces	20
About File Systems and Devices in Zones	27
Configurable Privileges	32
Setting Zone-Wide Resource Controls	33
Zone Configuration Data	35
Zone Global Properties	35
autoboot Global Property	36
autoshutdown Global Property	36
bootargs Global Property	36
brand Global Property	36
boot-priority Global Property	37
boot-disk-protection Global Property (solaris-kz Only)	37
cpu-arch Global Property (solaris-kz Only)	37
cpu-shares Global Property	39
description Global Property	39

file-mac-profile Global Property	39
fs-allowed Global Property (solaris and solaris10 Only)	40
global-time Global Property (solaris and solaris10 Only)	40
hostid Global Property	41
host-compatible Global Property	41
hwprovider Global Property (solaris10 Only)	43
ip-type Global Property	43
limitpriv Global Property	43
max-adi-metadata-memory Global Property	44
max-lofi Global Property	44
max-lwps Global Property	44
max-processes Global Property	44
max-System-V-resource Global Properties	44
pool Global Property	45
scheduling-class Global Property	45
tenant Global Property	45
zonename Global Property	46
zonepath Global Property (solaris Only)	46
Zone Resource Types and Their Properties	46
admin Resource Type	47
anet Resource Type	48
attr Resource Type	55
capped-cpu Resource Type	55
capped-memory Resource Type	56
dataset Resource Type (solaris and solaris10 Only)	57
dedicated-cpu Resource Type	59
device Resource Type	61
fs Resource Type	63
ib-vhca Resource Type (solaris-kz Only)	64
keysource Resource Type	66
net Resource Type	66
npiv Resource Type	67
rctl Resource Type	68
rootzpool Resource Type (solaris and solaris10 Only)	69
smf-dependency Resource Type	72

verified-boot Resource Type	72
virtual-cpu Resource Type (solaris-kz Only)	74
zpool Resource Type (solaris and solaris10 Only)	75
A Resource Types and Global Properties That Support Live Zone Reconfiguration	77
Live Zone Reconfiguration Support or Restriction	77
Index	79

Using This Documentation

- **Overview** – Describes how to configure Oracle Solaris zone resources.
- **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=E37838-01>.

Feedback

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

Non-Global Zone Configuration Command and Resources

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

This chapter covers the following topics:

- [“zonecfg Command” on page 11](#)
- [“About Resources in Zones” on page 15](#)
- [“Configurable Resource Types and Global Properties” on page 15](#)
- [“Zone Configuration Data” on page 35](#)
- [“Zone Global Properties” on page 35](#)
- [“Zone Resource Types and Their Properties” on page 46](#)

For further reference, review:

- [“About Zone Commands for Configuration and Installation” in *Creating and Using Oracle Solaris Zones*](#)
- [Unresolved link to " zonecfg Utility Edit Modes in Creating and Using Oracle Solaris Kernel Zones"](#)
- [Table 3, “Live Zone Reconfiguration Support or Restriction for Resource Types and Global Properties,” on page 77](#)

In Oracle Solaris Zones, the default branded zone is the `solaris` zone.

zonecfg Command

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 and verify the configuration.

The check performed by `zonecfg` for a given configuration verifies the following:

- Any required properties for a resource are specified.
- The configuration is free from conflicts.

For example, if a `zpool` resource type with a given `storage` property already exists, there will be a conflict on the name of the `zpool`. An `invalid-syntax` error will be displayed. If you have an `anet` resource, the zone is an `exclusive-IP` type and cannot be a `shared-IP` zone. Additionally, the `zonecfg` command displays a warning if an aliased dataset has a potential conflict with devices.

- No invalid parameter is specified.

If an invalid parameter is specified in the `zonecfg` command for a zone, the zone is not configured.

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 name is also included in the prompt:

```
zonecfg:zonename:fs>
```

For more information about the `zonecfg` command, see the [zonecfg\(8\)](#) man page.

zonecfg and the Global Zone

The `zonecfg` command can also be used to persistently specify resources for the global zone. For example, you can use the command to configure the global zone to use a dedicated CPU by setting a value for the `dedicated-cpu` resource type.

For more information, including procedures that show how to use the zonecfg command, see [Chapter 2, “Setting Up a Non-Global Zone”](#) in *Creating and Using Oracle Solaris Zones*.

zonecfg Templates

You can use different zonecfg templates to get a specific brand, to get an empty configuration, or to create a zone configuration similar to another zone that is already 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. Tokens in the template property can provide these unique values.

You can configure a solaris-kz branded zone by using the following templates:

- SYSsolaris-kz template – The default zone template configures a zone with 4 virtual CPUs and 4GB of memory.
- SYSsolaris-kz-minimal template – Configures a minimal kernel zone with 1 virtual CPU and 2GB of memory.

The zone remote administration daemon (RAD) module provides a systemic way to express, enforce, or implement changes by using the property templates. For more information, see the [zonemgr-1\(3rad\)](#) man page.

Tip - If the rad-zonemgr package was not initially installed on your system and you installed it later by using the `pkg install` command, you must use the `svcadm` command to restart `rad:local` and to restart `rad:remote` if it was running. Ensure that the RAD daemon loaded the module.

TABLE 1 zonecfg Template Tokens

Token	Description	Usage
{zonename}	The name of the zone.	Can be used by zonecfg as input from the user, or input from a template value.
{id}	A unique instance number that is the resource type's id property value.	Can be used by zonecfg as input from the user, or input from a template value. Evaluates to the id property of a particular resource type. Use within a resource scope that supports the id property.

Token	Description	Usage
{global-rootzpool}	Evaluates to the name of the root pool in the global zone.	Used in the default solaris-kz device resource type.
%	Evaluates to %.	Can be used by zonecfg as input from the user.

TABLE 2 Supported Tokens by Resource Type Property or Global Property

Resource Type	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} Token in the Zonepath

This example demonstrates how the {zonename} token enables the zonepath to be automatically updated when the zone name is changed.

```
zonecfg:ngz> info zonepath
zonepath.template: /system/zones/{zonename}
zonepath: /system/zones/ngz
```

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

EXAMPLE 2 Token Used for the storage Property of the device Resource Type for a Kernel Zone

```
device 0:
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
```

zoncfg Command-Line Editing

The Tecla command-line editing library is available to the `zoncfg` command. The library provides a mechanism for command-line history and editing support.

For more information, see the [tecla\(7\)](#) man page.

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.

Configurable Resource Types and Global Properties

This section provides overviews of some of the configurable zone resource types and global properties. Only the zone name is required. Additional information is provided in [“Zone Configuration Data” on page 35](#).

For information about configuration options that are specific to a particular brand of zone, see the [solaris\(7\)](#) and [solaris-kz\(7\)](#) man pages.

This section covers the following topics:

- [“Scheduling Class” on page 16](#)
- [“Capped Memory and Physical Memory Control” on page 16](#)
- [“Kernel Zone Migration Class and Host Compatibility Level \(solaris-kz Only\)” on page 18](#)
- [“About Zone Network Interfaces” on page 20](#)

- [“About File Systems and Devices in Zones” on page 27](#)
- [“Configurable Privileges” on page 32](#)
- [“Setting Zone-Wide Resource Controls” on page 33](#)

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.

The following actions are ways to set the scheduling class for a zone:

- Use the `scheduling-class` global property in `zonecfg` to set the scheduling class for the zone.
- 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 11.4*](#) and [“How to Associate a Pool With a Scheduling Class” in *Administering Resource Management in Oracle Solaris 11.4*](#).
- If the `cpu-shares` resource control is set and FSS has not been set as the scheduling class for the zone through another action, the `zoneadm` daemon 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. You use the `dispadm` command to set FSS to be the system default scheduling class.

You can use the `prionctl` command to move running processes into a different scheduling class without changing the default scheduling class and rebooting. For more information, see the [`prionctl\(1\)`](#) man page.

Capped Memory and Physical Memory Control

To use the `capped-memory` resource type, the `resource-cap` package must be installed in the global zone. This resource type is available to `solaris` zones and `solaris-kz` zones.

For more information about using capped memory, see the following documentation:

- Chapter 10, “About Controlling Physical Memory With the Resource Capping Daemon (rcapd)” in *Administering Resource Management in Oracle Solaris 11.4*
- Chapter 11, “Administering the Resource Capping Daemon Tasks” in *Administering Resource Management in Oracle Solaris 11.4*
- “How to Create and Deploy a Non-Global Zone” in *Creating and Using Oracle Solaris Zones*
- “About Memory Page Size Policy and Physical Memory” in *Creating and Using Oracle Solaris Kernel Zones*

Capped Memory Guidelines for a solaris Zone

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

- Determine values for the physical property of the capped-memory resource type by using the rcapd daemon from the global zone. The physical property is used by rcapd as the max-rss value for the zone.

The physical property of the capped-memory resource type represents a soft RAM allocation limit that is enforced by the rcapd daemon. 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 is 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 the swap property for a capped-memory resource without also setting the physical property can be an effective way to limit the amount of memory used by a non-global 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 type is the preferred way to set the zone.max-swap resource control for a non-global zone.
- The locked property of the capped-memory resource type is the preferred way to set the zone.max-locked-memory resource control for a non-global 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.

To temporarily set a resource cap for a zone, see [“Example: How to Specify a Temporary Resource Cap for a Zone”](#) in *Administering Resource Management in Oracle Solaris 11.4*.

For details about the capped-memory resource type, see [“capped-memory Resource Type”](#) on page 56 and the `solaris(7)` man page.

Capped Memory Guidelines for a `solaris-kz` Zone

For kernel zones, the `physical` property of the capped-memory resource type 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.

Values set for the `physical` property determine whether booting is successful. If a system supports 1-Gbyte pages, but the `physical` property is set to 2M, the value for `physical` is used even though the host supports a larger page size. Thus, booting with `pagesize-policy=largest-available` succeeds.

To determine the page sizes available on a system, type the following command:

```
global$ pagesize -a
```

The default `SYSsolaris-kz` template sets the `pagesize-policy` to `largest-available`, which is the recommended value. For kernel zones, the swap and locked limits are not allowed. The `rcapd` utility is not used. Live Zone Reconfiguration is not supported.

To determine system support for the `pagesize-policy` property, use the following command:

```
global$ zoneadm migrate -n
```

For details about the capped-memory resource type, see [“capped-memory Resource Type”](#) on page 56 and the `solaris-kz(7)` man page.

Kernel Zone Migration Class and Host Compatibility Level (`solaris-kz` Only)

Only features enabled by both migration class (`cpu-arch`) and host compatibility level (`host-compatible`) are visible to a SPARC kernel zone. To migrate a SPARC 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 `host-compatible` global properties.

If not set, the default value of `cpu-arch` is `solaris`. 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.

Do not set the `cpu-arch` global property to a migration class if you want to use SSM or DAX.

This section provides the following additional overview information:

- [“Cross-CPU Migration Classes \(`solaris-kz`\)” on page 19](#)
- [“Cross-CPU Migration Classes \(`solaris-kz`\)” on page 20](#)
- [“Host Compatibility Levels \(`solaris-kz` SPARC Only\)” on page 20](#)

For details about configuring CPU migration classes and host compatibility levels for kernel zones, see [“`cpu-arch` Global Property \(`solaris-kz` Only\)” on page 37](#), [“Host Compatibility Levels \(`solaris-kz` SPARC Only\)” on page 20](#), and the `solaris-kz(7)` man page.

SPARC: Cross-CPU Migration Classes (`solaris-kz`)

Use the `cpu-arch` global property to configure SPARC 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.

The default value of the `cpu-arch` global property is `solaris`. By default, 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 system will not boot if the `cpu-arch` global property 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, Fujitsu SPARC M12, and Fujitsu M10/SPARC M10 supported systems.

For configuration details, see [“`cpu-arch` Global Property \(`solaris-kz` Only\)” on page 37](#) and the `solaris-kz(7)` man page.

x86: Cross-CPU Migration Classes (solaris-kz)

Only features enabled by the migration class are visible to an x86 kernel zone. To migrate an x86 kernel zone, 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` global property. The source and the target hosts must be the same platform. On x86 hosts, the CPU model names must match.

If no value is set, the default value of `cpu-arch` is `solaris`. The zone boots with the same CPU class as the host. You can migrate the zone between CPU types that are of exactly the same type or are the same micro architecture.

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. The CPU class of the zone cannot exceed the limits of the CPU class of the host.

For configuration details, see [“cpu-arch Global Property \(solaris-kz Only\)” on page 37](#) and the [solaris-kz\(7\)](#) man page.

Host Compatibility Levels (solaris-kz SPARC Only)

Use the `host-compatibility` global property to enable compatible features between the source and target hosts. The compatible features are those features supported by the version of Oracle Solaris running in the global zone. For live or warm migration between hosts running different Oracle Solaris releases, the `host-compatibility` global property enables you to control the features that can be used by the zone.

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

For configuration details, see [“Host Compatibility Levels \(solaris-kz SPARC Only\)” on page 20](#) and the [solaris-kz\(7\)](#) man page.

About Zone Network Interfaces

This section provides the following information:

- [“Guidelines for Network Interfaces in Zones” on page 21](#)
- [“About Datalinks and Zones” on page 22](#)
- [“About Elastic Virtual Switch and Zones” on page 22](#)

- [“Exclusive-IP Non-Global Zones” on page 23](#)
- [“Shared-IP Non-Global Zones” on page 25](#)
- [“Reliable Datagram Sockets Support in Non-Global Zones” on page 26](#)
- [“Datalink Security Differences Between Shared-IP and Exclusive-IP Non-Global Zones” on page 27](#)

Guidelines for Network Interfaces in Zones

This section provides guidelines for network interfaces used by zones.

- **Exclusive-IP network configuration** – 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 by using the `anet` resource type, a dedicated VNIC is automatically created and assigned to that zone. By using the automated `anet` resource type, the requirement to create and configure datalinks in the global zone and assign the datalinks to non-global zones is eliminated. Use the `anet` resource type to accomplish the following:
 - Allow the global zone administrator to choose specific names for the datalinks assigned to non-global zones
 - Allow multiple zones to use datalinks of the same name

For additional information about IP features in each type for exclusive-IP zones, see [“Exclusive-IP Non-Global Zones” on page 23](#) and [“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*](#).

- **SMF management of network configuration** – To manage the persistent network configuration, use the Service Management Facility (SMF). Network configuration is aligned with other system components that use SMF as a core storage repository. Network configuration through SMF also enables you to perform customized network configuration during an installation by specifying datalink, flow, and IP properties in a system configuration profile. The properties are administered by the `dladm`, `flowadm`, and `ipadm` commands. For details about the `dladm` property names that support SMF, see [Configuring and Managing Network Components in Oracle Solaris 11.4](#).
- **Network set-up at boot** – 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.
- **IP layers** – 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 Packet Filter.

- **Multiple network resources** – 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 uses two anet resources. The first automatically configures the 192.168.3.3 address on one of the zone's interfaces. The second allows 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>
```

- **Preconfigured datalinks** – For backward compatibility, preconfigured datalinks can be assigned to non-global zones.
- **Link protection in zones** – The link protection described in [Chapter 1, “Using Link Protection in Virtualized Environments”](#) in *Securing the Network in Oracle Solaris 11.4* can be used on a system running zones. This functionality is configured in the global zone.

About Datalinks and Zones

A datalink 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 datalink is also referred to as a physical interface, for example, a Network Interface Card (NIC). The datalink is the physical property configured by using the zonecfg utility. The physical property can be a VNIC.

By default, 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 resource type description in [“Zone Resource Types and Their Properties”](#) on page 46.

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	max-bw
defrouter	mtu
lower-link	priority
mac-address	vlan-id

You can set the following properties for an EVS anet resource:

configure-allowed-address	linkname
evs	vport

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

The following example input for a zone named `evszone` sets the tenant global property for a tenant named `tenantA`. The anet resource type 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:my-zone> set tenant=tenantA
zonecfg:my-zone> add anet
zonecfg:my-zone> set evs=EVSA
zonecfg:my-zone> set vport=vport0
```

For more information, see [Chapter 5, “About Elastic Virtual Switches” in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*](#).

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 datalinks.

The following features can be used in an exclusive-IP zone:

- DHCPv4 and IPv6 stateless address autoconfiguration
- Packet 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 type 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 datalink in the global zone and assigned it to the exclusive-IP zone by using the `net` resource type of the `zonecfg` utility. The datalink is specified by using the `physical` property of the `net` resource type. The `physical` property can be a VNIC. The `address` property of the `net` resource type is not set.

Note - An assigned datalink 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, you can specify a comma-separated list of IP addresses by using the `allowed-address` property of the `anet` resource type. 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.

In the following 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 datalinks to running exclusive-IP zones. The `dladm` command can be used with the `set-linkprop` subcommand to assign additional datalinks to running zones. See [“Managing Datalinks Related to Zones” in *Creating and Using Oracle Solaris Zones*](#) for usage examples.

Inside a running exclusive-IP zone that is assigned its own set of datalinks, 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(8)` man page.

The IP configuration of an exclusive-IP zone can only be viewed from the global zone by using the `zlogin` command, as shown in the following example.

```
global$ zlogin my-zone 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
```

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`.

```
global$ 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*](#).

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 you log in to the zone, the `rds` service can be enabled in each zone in which it is to run.

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

1. Become a zone administrator.

For more information, see [“Using Rights Profiles to Install and Manage Zones” in *Creating and Using Oracle Solaris Zones*](#).

2. Log in to the zone with the `zlogin` command.

3. Enable the RDSv3 service in a zone.

```
my-zone$ pfbash svcadm enable rds
```

For more information, see the [`svcadm\(8\)`](#) man page.

4. Verify that the `rds` service is enabled.

```
my-zone$ svcs rds
STATE          STIME          FMRI
online         22:50:53      svc:/system/rds:default
```

Datalink 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, the `zonecfg` command grants the entire specified datalink 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 datalinks, 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 type, 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.

About File Systems and Devices in Zones

This section provides the following information:

- [“File Systems Mounted in Zones” on page 27](#)
- [“File System Mounts and Updating” on page 29](#)
- [“/dev File System in Non-Global Zones” on page 29](#)
- [“Removable `lofi` Device in Non-Global Zones” on page 29](#)
- [“Disk Format Support in Non-Global Zones” on page 30](#)
- [“Kernel Zones Device Resources With Storage URIs” on page 30](#)

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.

Note - Do not manipulate the `.../rpool/ROOT` dataset 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.

Note - Do *not* use the `zfs` command 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 the [sharectl\(8\)](#) man page.
- Mounts explicitly performed by a zone administrator

File system mounting permissions within a running non-global zone are also defined by the `fs-allowed` global property. This property does not apply to file systems mounted into the zone by using the `zonecfg add fs` or `add dataset` commands. 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\(7\)](#) man page. For more information about BEs, see [Creating and Administering Oracle Solaris 11.4 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\(7\)](#) 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 `rpoul/export` dataset or another delegated dataset.

/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 Create and Deploy a Non-Global 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. 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.

EXAMPLE 3 Using the `lofiadm` Command to List Removable `lofi` Devices

The following command creates a removable `lofi` device with an associated file.

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

The following command creates an empty removable `lofi` device.

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

The following command inserts a file into a removable `lofi` device.

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

For more information, see the [lofiadm\(8\)](#), [zonecfg\(8\)](#), and [lofi\(4D\)](#) man pages. Also see “Setting Zone-Wide Resource Controls” on page 33.

Disk Format Support in Non-Global Zones

Disk partitioning and use of the `uscsi` command are enabled through the `zonecfg` tool. See device in “Zone Resource Types and Their Properties” on page 46 for an example. For more information about the `uscsi` command, see [uscsi\(4I\)](#).

- Delegation is only supported for `solaris` zones.
- Disks must use the `sd` target as shown by using the `prtconf -D` command. See the [prtconf\(8\)](#) man page.

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 the `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 usernames 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 in device resources.
 - Only set the `bootpri` property 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 might damage the data on the disk.
 - Only set the `bootpri` property on devices that must be bootable.
 - The `id` property controls the instance of the disk in the kernel zone. For example, `id=5` means that the disk will be `c1d5` in the zone.

For more information about the `bootpri` and `id` properties, see the [solaris-kz\(7\)](#) man page.

- The root `zpool` that is created on bootable `solaris-kz` disks can be imported into the global zone during installation. The root `zpool` is visible with the `zpool` command. See the [zpool\(8\)](#) man page for more information.

EXAMPLE 4 Configuring a Storage URI to Create a Portable Zone Configuration

This example uses a device resource type to configure a storage URI that makes the zone configuration portable to other host systems.

```
$ pfbash 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\(7\)](#) man page.

EXAMPLE 5 Viewing the Current Device Resources Configuration

This example displays information about the current configuration for device resources.

```
$ pfbash zonecfg -z my-zone info device
device:
  storage: dev:/dev/zvol/dsk/rpool/VARSHARE/zones/my-zone/disk0
  id: 0
  bootpri: 0
device:
  storage: nfs://user1:staff@host1/export/file1
  create-size: 4g
```

EXAMPLE 6 Viewing the Current Device Resources Configuration for a Specified ID

This example displays the output for a specific zone by specifying the ID for the zone.

```
$ zonecfg -z my-zone info device id=1
device:
  storage: nfs://user1:staff@host1/export/file1
  create-size: 4g
  id: 1
  bootpri not specified
```

Configurable Privileges

When a zone is booted, a default set of *safe* privileges is included in the configuration. These privileges are considered safe because they prevent a privileged process in the zone from affecting processes in other non-global zones on the system or in the global zone. You can use the `zonecfg` command to do the following:

- Add to the default set of privileges, understanding that such changes might allow processes in one zone to affect processes in other zones by being able to control a global resource.
- Remove from the default set of privileges, understanding that such changes might prevent some processes from operating correctly if they require those privileges to run.

Note - A few privileges cannot be removed from the zone's default privilege set or cannot be added to the set.

For more information, review the following:

- [“Privileges in a Non-Global Zone”](#) in *Creating and Using Oracle Solaris Zones*
- [“How to Create and Deploy a Non-Global Zone”](#) in *Creating and Using Oracle Solaris Zones*
- [“Modifying Which Privileges Are Available on a System”](#) in *Securing Users and Processes in Oracle Solaris 11.4*
- [privileges\(7\)](#) man page

Setting Zone-Wide Resource Controls

Resource controls are specified for both the global and non-global zones by using the `zonecfg` command.

- Zone-wide resource controls limit the total resource usage of all process entities within a zone. For a complete list, see the [resource-controls\(7\)](#) man page. For zone-specific resources, search for "zone" in the man page.
- An administrator with the appropriate rights can set zone-wide resource controls for a zone. For information about rights, see [“Using Rights Profiles to Install and Manage Zones” in *Creating and Using Oracle Solaris Zones*](#).

Tip - For ease of administration, set the resource control as a zone resource property, such as `capped-cpu`, rather than the equivalent resource control, such as `zone.cpu-cap`.

- `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.
See also [“capped-cpu Resource Type” on page 55](#).
- `zone.cpu-shares` **resource control** – Global scope. 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 a System With Zones Installed” in *Creating and Using Oracle Solaris Zones*](#). The global property name for this control is `cpu-shares`.
See also [“cpu-shares Global Property” on page 39](#).
- `zone.max-adi-metadata-memory` **resource control** – Global scope. Controls the maximum amount of metadata allocated for Silicon Secured Memory (SSM) enabled pageable memory. SSM includes ADI.
See also [“max-adi-metadata-memory Global Property” on page 44](#).
- `zone.max-locked-memory` **resource control** – Limits the amount of locked physical memory available to a non-global 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.

Tip - 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.

For more information, see [“Available Resource Controls” in *Administering Resource Management in Oracle Solaris 11.4*](#) and the locked property in [“capped-memory Resource Type” on page 56](#).

- `zone.max-lofi` **resource control** – Limits the number of potential `lofi` devices that can be created by a zone.

See also [“max-lofi Global Property” on page 44](#).

- `zone.max-lwps` **resource control** – Global scope. 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 11.4*](#) for more information.

See also [“max-lwps Global Property” on page 44](#).

- `zone.max-processes` **resource control** – Global scope. 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 11.4*](#). 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 - Because both normal processes and zombie processes take up process table slots, the `max-processes` control thus protects against zombie processes exhausting the process table. Because zombie processes do not have any LWPs by definition, the `max-lwps` cannot protect against this possibility.

See also [“max-processes Global Property” on page 44](#).

- `zone.max-msg-ids` **resource control** – Global scope. And `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 equivalent global property names for these controls are `max-msg-ids`, `max-sem-ids`, `max-shm-ids`, and `max-shm-memory`, respectively.
- `zone.max-swap` **resource control** – Limits swap consumed by user process address space mappings and `tmpfs` mounts within a zone. The output of the `prstat -Z` command displays a SWAP column. The swap reported is the total swap consumed by the processes of the

zone 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.

See also the swap property in [“capped-memory Resource Type” on page 56](#).

You can specify these resource controls on running processes by using the `prctl` command. An example is provided in [“How to Temporarily Set FSS Shares in the Global Zone” in *Creating and Using Oracle Solaris Zones*](#). However, controls set by using `prctl` do not persist across system reboot.

Zone Configuration Data

Zone configuration data consists of two kinds of entities: global properties and resource types.

- Global properties are properties of the configuration as a whole, rather than of a particular resource.

See [“Zone Global Properties” on page 35](#) for details.

- Each zone resource type can have a set of one or more resource type properties. The resource type properties have names and values.

See [“Zone Resource Types and Their Properties” on page 46](#) for details.

For a complete listing and descriptions of zone resource types and global properties, see the [`zonecfg\(8\)`](#) man page.

Zone Global Properties

The following sections describe the global properties of a zone. Global properties apply to the zone configuration as a whole. For the complete list of global properties and their descriptions, see the [`zonecfg\(8\)`](#) man page.

For information about which global properties support Live Zone Reconfiguration in `solaris` and kernel zones, see [Appendix A, “Resource Types and Global Properties That Support Live Zone Reconfiguration”](#).

autoboot Global Property

Sets whether the zone is automatically booted when the global zone is booted. The zones service, `svc:/system/zones:default`, must also be enabled.

This property is set to `false` by default. If it is set to `true`, the zone is automatically booted when the global zone is booted.

Note - 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:

```
global$ svcadm enable zones
```

autosutdown Global Property

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

bootargs Global Property

Sets a boot argument for the zone.

The boot argument is applied unless overridden by the `reboot`, `zoneadm boot`, or `zoneadm reboot` commands.

For the list of valid arguments, see the [zoneadm\(8\)](#) man page under "`zoneadm boot`".

brand Global Property

Indicates the brand of the zone, one of `labeled`, `solaris`, `solaris10`, and `solaris-kz`. For more information, see the [brands\(7\)](#) man page.

boot-priority Global Property

Sets the booting order of zones on a system. Once the SMF dependencies are satisfied for a zone, the zone boots according to its configured `boot-priority`. Values for `boot-priority` are `high`, `normal`, and `low`. The default value is `normal`.

boot-disk-protection Global Property (solaris-kz Only)

Enables or disables boot disk protection for a `solaris-kz` zone. Values are `on` and `off`. The default value is `off`. This feature requires that all boot disks be on storage LUNs that support SCSI-3 PGR reservation.

cpu-arch Global Property (solaris-kz Only)

Specifies the migration class configured for a `solaris-kz` zone.

Only features enabled by the specified migration class are visible to an x86 kernel zone. To migrate an x86 kernel zone, 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` global property. The source and the target hosts must be the same platform. On x86 hosts, the CPU model names must match.

If no value is set, the default value of `cpu-arch` is `solaris`. The zone boots with the same CPU class as the host. You can migrate the zone between CPU types that are of exactly the same type or are the same micro architecture.

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. The CPU class of the zone cannot exceed the limits of the CPU class of the host.

Note - Performance counters are not available when `cpu-arch` is set to a migration class.

No migration classes apply to AMD CPUs only.

The migration classes are as follows:

`generic` (SPARC only)

Kernel zone can perform a CPU-type independent migration, but not to a system older than the SPARC T4.

Use the `generic` class to migrate kernel zones between Oracle SPARC systems and Fujitsu systems.

`migration-class1`

Kernel zone can perform cross-CPU type migration between CPUs of Nehalem or later micro architectures.

This class supports the following features:

<code>ahf64</code>	<code>fpu</code>	<code>mmx</code>	<code>pse</code>	<code>sse4.1</code>
<code>ahf64</code>	<code>fxsr</code>	<code>msr</code>	<code>pse36</code>	<code>sse4.2</code>
<code>clflush</code>	<code>pge</code>	<code>nx-bit</code>	<code>ss</code>	<code>ssse</code>
<code>cmov</code>	<code>htt</code>	<code>pae</code>	<code>sse</code>	<code>sysc</code>
<code>cx8</code>	<code>long-mode</code>	<code>pdcn</code>	<code>sse2</code>	<code>tsc</code>
<code>cx16</code>	<code>mce</code>	<code>popcnt</code>	<code>sse3</code>	<code>tscp</code>

`migration-class2` (x86 only)

Kernel zone can perform cross-CPU type migration between CPUs of Westmere or later micro architectures.

Features supported by this class are all features supported by `migration-class1` plus `aes`, `lg-page`, and `pclmulqdq`.

`migration-class3` (x86 only)

Kernel zone can perform cross-CPU type migration between CPUs of Sandy Bridge or later micro architectures.

Features supported by this class are all features supported by `migration-class2` plus `avx` and `xsave`.

`migration-class4` (x86 only)

Kernel zone can perform cross-CPU type migration between CPUs of Ivy Bridge or later micro architectures.

Features supported by this class are all features supported by `migration-class3` plus `efs`, `f16c`, and `rdrand`.

`migration-class5` (x86 only)

Kernel zone can perform cross-CPU type migration between CPUs of Haswell or later micro architectures.

Features supported by this class are all features supported by `migration-class4` plus `avx2`, `bmi1`, `bmi2`, `fma`, `lzcnt`, and `movbe`.

`migration-class6` (x86 only)

Kernel zone can perform cross-CPU type migration between CPUs of Haswell or later micro architectures.

Features supported by this class are all features supported by `migration-class5` plus `adx`, `prfchw`, and `rdseed`

`sparc64-class1` (SPARC only)

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

If no value is set, the kernel zone can migrate between CPUs of the same micro architecture or of the exact same type, if the micro architecture cannot be recognized.

cpu-shares Global Property

Sets a limit on the number of fair share scheduler (FSS) CPU shares for a zone. For more information, see [“Scheduling Class” on page 16](#) and [“Using the Fair Share Scheduler on a System With Zones Installed” in *Creating and Using Oracle Solaris Zones*](#).

description Global Property

Specifies an optional description of the zone. The `description` property value is a string of up to 255 US-ASCII characters.

Use this property when you want to provide a more detailed description of the zone and what it does.

file-mac-profile Global Property

Configures Immutable Zones. The possible values in order of which public files can be modified are:

none
flexible-configuration
dynamic-zones (global zone only)
fixed-configuration
strict

For more information, see [Chapter 10, “Configuring and Administering Immutable Zones”](#) in *Creating and Using Oracle Solaris Zones*.

fs-allowed Global Property (solaris and solaris10 Only)

Enables 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` global property. By default, only mounts of `hfs` 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` global 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` commands.

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*.

global-time Global Property (solaris and solaris10 Only)

Enables either the system-wide or zone-specific time to be set from within the non-global zone:

- `global-time=true` indicates that the zone is allowed to set system-wide time.

- `global-time=false` indicates the zone is allowed to set zone-specific time.

You should assign a value for the `global-time` global property. However, if the value is not set but the `sys_time` privilege is explicitly assigned using the `limitpriv` property, the value of `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` global property setting in Oracle Solaris, a process within a non-global zone can manipulate either the virtual zone-specific time or the system-wide time by using the following system calls. The process must have the `sys_time` privilege.

- `stime`
- `clock_gettime`
- 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 about privileges.

hostid Global Property

Sets a `hostid` global 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 global zone that is migrated into a zone on another system. Applications now inside the zone might depend on the original `hostid`.

host-compatible Global Property

Specifies the host compatibility level configured for a `solaris-kz` brand zone.

A compatibility level is used to enable features on a target host during live or warm migration.

Only features enabled by both migration class and host compatibility level are visible to the kernel zone.

Features included in a compatibility level can be extended by specifying compatibility level modifiers. A modifier can only be used with designated compatibility level as listed after each modifier. The modifiers work on the SPARC platform only.

The possible host compatibility levels are as follows:

`solaris` Make available all the features in the current version of Oracle Solaris.

Note - The `solaris` host compatibility level might prevent the kernel zone from being migrated to a host running a different release of Oracle Solaris.

`level1` If all of your systems are running the Oracle Solaris 11.4 release, set the `host-compatible` property to `level1`, which allows enabling of all features available in the release. The `level1` level includes Silicon Secured Memory (SSM) or ADI, SPARC M7 and later SPARC series systems DAX Data Analytics Accelerator (DAX) coprocessors, and VA Mask features.

On DAX-capable hardware, DAX is always turned on in the global zone. For information about using DAX to configure the Oracle Database 12c in-memory feature, see [Using the In-Memory Column Store](#).

Note - The `level1` setting might prevent the kernel zone from being migrated to other hosts that are running an older release of Oracle Solaris.

The SPARC modifier is as follows:

`adi` Enable the Silicon Secured Memory (SSM) feature, also known as ADI. By default, SSM is turned off for a kernel zone. On SSM-capable hardware, SSM is always turned on in the global zone. The `host-compatible` global property cannot be used to enable SSM if the SSM feature is not supported by the migration class.

The `adi` modifier can only be used with the default compatibility level.

Note - Use of the `adi` setting might prevent the kernel zone from being migrated to other hosts that are running an older release of Oracle Solaris.

For more information about Oracle Solaris and ADI or silicon secured memory, refer to the [adi\(2\)](#), [memcntl\(2\)](#), [mmap\(2\)](#), [adi\(3C\)](#), and [siginfo\(3HEAD\)](#) man pages. For more information about SSM functionality, visit [Introduction to SPARC M7 and Silicon Secured Memory \(SSM\)](#).

If no value is set, the default kernel zone's host compatibility level will only include features supported in the source host.

hwprovider Global Property (solaris10 Only)

Configures the hardware manufacturer string for a `solaris10` branded zone. The default configuration in a zone is no value. You might need to set the value to `Sun_Microsystems` when you are moving this zone to an Oracle Solaris 11.4 system.

For instructions, see [“Setting the hwprovider Property to Override the Global Zone Value \(solaris10 Only\)”](#) in *Creating and Using Oracle Solaris Zones*.

When this property is untouched or cleared, the hardware manufacturer string of the global zone is used for the `solaris10` branded zone. See the [`sysinfo\(2\)`](#) man page.

ip-type Global Property

Required for all non-global zones. Values are one of `exclusive` and `shared`. See [“Exclusive-IP Non-Global Zones”](#) on page 23, [“Shared-IP Non-Global Zones”](#) on page 25, and [“How to Create and Deploy a Non-Global Zone”](#) in *Creating and Using Oracle Solaris Zones*.

limitpriv Global Property

Specifies 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 example entry adds the ability to use DTrace programs that only require the `dttrace_proc` and `dttrace_user` privileges in the zone:

```
global$ pfexec zonecfg -z my-zone
zonecfg:my-zone> set limitpriv="default,dttrace_proc,dttrace_user"
```

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

```
global$ pfexec zonecfg -z my-zone
zonecfg:my-zone> set limitpriv="default,proc_pricntl"
```

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.

max-adi-metadata-memory Global Property

Sets the maximum number of process table slots that are simultaneously available to a zone.

max-lofi Global Property

Sets the maximum number of the loopback interfaces for a zone.

max-lwps Global Property

Sets the maximum number of the LWPs that are simultaneously available to a zone.

max-processes Global Property

Sets the maximum number of process table slots that are simultaneously available to a zone.

max-System-V-resource Global Properties

Sets the maximum number of the *System-V-resource*. The following is the complete list of *max-System-V-resource* global properties:

max-msg-ids max-semids max-shm-ids max-shm-memory

For descriptions and information about the effects of setting a maximum System V property value on another property, see [“Setting Zone-Wide Resource Controls” on page 33](#) and the `zonecfg(8)` man page.

pool Global Property

Associates the zone with a resource pool on the system. Multiple zones can share the resources of one pool.

If you have configured resource pools on your system as described in [Chapter 13, “Creating and Administering Resource Pools Tasks” in *Administering Resource Management in Oracle Solaris 11.4*](#), you can use the `pool` global property to associate the zone with one of the resource pools when you configure the zone.

The `pool` global 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` global property is incompatible with a temporary pool configured through the `dedicated-cpu` resource type. You can set only one of these two properties.

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 type. 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 type.

scheduling-class Global Property

Sets the scheduling class for the zone. See [“Scheduling Class” on page 16](#) for more information and additional ways to set the scheduling class.

tenant Global Property

Manages the namespace for EVS tenants. EVS resources that are defined within a tenant global property are not visible outside that tenant's namespace. For more information and an example, see [“About Elastic Virtual Switch and Zones” on page 22](#).

zonename Global Property

Required for all zones. Sets the name of the zone. The following rules apply to zone names:

- Each zone must have a unique name.
- A zone name is case-sensitive.
- A zone name must begin with an alphanumeric character.
 - The name can contain alphanumeric characters, underbars (`_`), hyphens (`-`), and periods (`.`).
- The name cannot be longer than 63 characters.
- The name `global` is reserved for the global zone.
- Names beginning with `SYS` are reserved and cannot be used.

zonepath Global Property (solaris Only)

In `solaris` zones created with the `zonecfg template` property, the default value of `zonepath` is `/system/zones/%{zonename}`.

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

Note - You can move a zone to another location on the same system by specifying a new, full `zonepath` by using the `zoneadm move` command. See [“How to Move a Zone That Is Not on Shared Storage”](#) in *Creating and Using Oracle Solaris Zones* for instructions.

Zone Resource Types and Their Properties

The following sections describe the resource types and resource type properties of a zone. See also the [`zonecfg\(8\)`](#) man page.

Tip - 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.

For information about which resource types and their properties support Live Zone Reconfiguration in non-global and kernel zones, see [Appendix A, “Resource Types and Global Properties That Support Live Zone Reconfiguration”](#).

admin Resource Type

The `admin` setting allows you to set zone administration authorization. Note that these auths do not allow you to create a zone. This capability is included in the Zone Security rights profile. For details about the Zone rights profiles, see [“Using Rights Profiles to Install and Manage Zones”](#) in *Creating and Using Oracle Solaris Zones*.

Properties: name, auths, user

user Property

Specify the user name.

auths Property

Specify the authorizations for the user name.

The values for the `auths` property are as follows:

`clone (solaris.zone.clonefrom)`

This authorization 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`.

`config (solaris.zone.config)`

This authorization allows modification of the persistent configuration of the zone by using the authorization `solaris.zone.config/zonename`.

`liveconfig (solaris.zone.liveconfig)`

This authorization allows inspection and modification of the live zone configuration by using the authorization `solaris.zone.liveconfig/zonename`. For more information about the live zone configuration, see [Unresolved link to " Live Zone Reconfiguration of Kernel Zones in Creating and Using Oracle Solaris Kernel Zones"](#) and [Unresolved link to " Live Zone Reconfiguration of Kernel Zones in Creating and Using Oracle Solaris Kernel Zones"](#).

`login (solaris.zone.login)`

This authorization 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 the [zlogin\(1\)](#) man page.

`manage (solaris.zone.manage)`

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

`migrate (solaris.zone.migrate)`

This authorization allows a user to perform all types of migration, cold, warm, and live, for a specific zone.

`migrate.cold (solaris.zone.migrate.cold)`

This authorization allows a user to do cold and warm migration of a specific zone.

Note - When migrating a kernel zone using authorizations set through the `admin` resource type on the source system, the zone configuration must be pre-defined on the target system for the migration to succeed.

For examples, see “Using Rights Profiles to Install and Manage Zones” in *Creating and Using Oracle Solaris Zones* and “Rights Required to Perform Kernel Zone Migrations” in *Creating and Using Oracle Solaris Kernel Zones*.

For more information about authorizations, see the [auths\(1\)](#), [auth_attr\(5\)](#), and [user_attr\(5\)](#) man pages.

anet Resource Type

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

Properties:

<code>allowed-address</code>	<code>id</code>	<code>pkey (IPoIB)</code>
<code>allowed-dhcp-cids</code>	<code>iov</code>	<code>priority</code>
<code>allowed-mac-address</code>	<code>linkmode (IPoIB)</code>	<code>ring-group</code>
<code>allowed-vlan-ids</code>	<code>link-protection</code>	<code>rxfanout</code>
<code>auto-mac-address</code>	<code>linkname</code>	<code>rxrings</code>
<code>autopush</code>	<code>lower-link (IPoIB)</code>	<code>txrings</code>
<code>bwshare</code>	<code>mac (solaris-kz only)</code>	<code>vlan (solaris-kz only)</code>

configure-allowed-address	mac-address (non-IPoIB)	vlan-id (non-IPoIB)
cos	mac-prefix (non-IPoIB)	vport
defrouter	mac-slot (non-IPoIB)	vsi-mgrid
etsbw-lcl	maxbw	vsi-typeid
evs	mtu	vsi-vers

For descriptions of these properties, see the [zonecfg\(8\)](#) man page.

For `solaris-kz` zones 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:

```
global$ 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 `allowed-mac-address` property of the `anet` resource type 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:my-kernelzone> add anet
zonecfg:my-kernelzone:anet> add mac
zonecfg:my-kernelzone:anet:mac> add allowed-mac-address fa:16:3f
zonecfg:my-kernelzone:anet:mac> add allowed-mac-address fa:80:20:21:22
zonecfg:my-kernelzone:anet:mac> end
zonecfg:my-kernelzone: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 type.

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

- The `allowed-vlan-ids` property of the `anet` resource type 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:my-kernelzone> add anet
```

```

zonecfg:my-kernelzone:anet> add vlan
zonecfg:my-kernelzone:anet:vlan> add allowed-vlan-ids 100-199
zonecfg:my-kernelzone:anet:vlan> add allowed-vlan-ids 400-498
zonecfg:my-kernelzone:anet:vlan> add allowed-vlan-ids 500
zonecfg:my-kernelzone:anet:vlan> end
zonecfg:my-kernelzone: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 type.

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

<code>allowed-address</code>	<code>priority</code>
<code>allowed-dhcp-cids</code>	<code>rx-fanout</code>
<code>configure-allowed-address</code>	<code>rx-rings</code>
<code>cos</code>	<code>vlan-id</code>
<code>defrouter</code>	<code>ring-group</code>
<code>etsbw-lcl</code>	<code>vport</code>
<code>evs</code>	<code>vsi-mgrid</code>
<code>link-protection</code>	<code>vsi-typeid</code>
<code>max-bw</code>	<code>vsi-vers</code>
<code>mtu</code>	

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(8)` man page.

Note - For kernel zone warm migrations, unless both the operating system instance and the kernel zone are running Oracle Solaris 11.4, suspend and resume operations are not supported if the `zonecfg iov` property is set to `auto` or `on`. For further information about kernel zone suspend and resume operations, see [“Using Warm Migration to Migrate a Kernel Zone”](#) in *Creating and Using Oracle Solaris Kernel Zones*.

For `solaris` zones only, do not use the `zonecfg` utility to set the following `anet` properties for IPoIB datalinks.

<code>mac-address</code>	<code>mac-prefix</code>	<code>mac-slot</code>	<code>vlan-id</code>
--------------------------	-------------------------	-----------------------	----------------------

For non-IPoIB datalinks, do not use the `zonecfg` utility to set the following `anet` properties.

link-mode
pkey

For an EVS anet resource, set only the following properties:

configure-allowed-address	linkname
evs	vport

The anet resource type 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 - The solaris-kz brand does not support IPoIB.

allowed-address Property

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.

autopush Property

Set to support the dladm link property autopush.

The following example automatically pushes the *ipsecah* streams module onto *net0* when the IP interface is plumbed inside the zone:

```
global$ pfbash zonecfg -z my-zone
zonecfg:my-zone> select anet linkname=net0
zonecfg:my-zone:anet> set autopush=ipsecah
zonecfg:my-zone:anet> end
zonecfg:my-zone> exit
```

If multiple modules are listed, they are pushed in the order specified.

```
zonecfg:my-zone:anet> set autopush="ipsecah ipsecesp"
```

defrouter Property

The defrouter property is 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.

etsbw-lcl Property

Indicates the ETS bandwidth on the TX side. See the [dladm\(8\)](#) man page for more information about this property.

`iov` Property

See “[Managing Single-Root I/O NIC Virtualization on Kernel Zones](#)” in *Creating and Using Oracle Solaris Kernel Zones*. For specific information about shadow VNICS used to provide network statistics, see “[Using Virtual Functions and Shadow VNICS With Kernel Zones](#)” in *Creating and Using Oracle Solaris Kernel Zones*.

`lower-link` Property

Specifies the underlying link for the link to be created. When set to `auto`, the `zoneadm` 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 `datalink` for creating the VNIC automatically during boot.

`link-mode` Property (**IPoIB only**)

Sets the `link-mode` for the `datalink` interface. The default value is `cm`. Valid values are as follows:

`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` Property

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

`mac-address` Property (**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\(8\)](#) 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 Property (IPoIB only)

Sets the partition key to be used for creating the IPoIB datalink interface. This property is mandatory. The specified pkey is always treated as hexadecimal, whether or not it has the 0x prefix.

ring-group Property

Set this property to allow a zone to use the hardware ring group capability of the Ethernet link. The possible values for this property are as follows:

auto

The system determines whether exclusive or shared is used on a particular lower-link. This is the default value.

exclusive

Use an exclusive hardware ring group. If an exclusive hardware ring group is not available, anet creation fails.

shared

Do not use a dedicated hardware ring group.

vlan-id Property (not for IPoIB)

Enable VLAN or PVLAN tagging for this VNIC and specify an ID for the VLAN tag. There is no default value. If this property is not set, the VNIC does not participate in any VLAN.

Note - The `vlan-id` property is not supported on IPoIB datalinks.

See the [dladm\(8\)](#) man page for the supported VLAN ID format.

When the `zonecfg` command creates a zone by 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 `netN`, `netX`. To change the default values, use the `zonecfg` command.

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

- Oracle Solaris zones – Random `mac-address`
- Oracle Solaris kernel zones – Random `mac-address`
- Oracle Solaris zones under a kernel zone – Factory `mac-address`
- Oracle VM Server for SPARC guest domains – Factory `mac-address`

- Oracle Solaris kernel zones running on an 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 - The `ip-type=exclusive` property 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 datalink 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
```

A kernel zone can serve as the global zone to `solaris` non-global zones. For information about managing `solaris` zones in kernel zones, see [“Managing Non-Global Zones in Kernel Zones”](#) in *Creating and Using Oracle Solaris Kernel Zones*, and [“How to Add Multiple MAC Addresses to a Kernel Zone”](#) in *Creating and Using Oracle Solaris Kernel Zones*.

For more information about properties, see the `zonecfg(8)` man page. For additional information about the link properties, see the `dladm(8)` man page.

attr Resource Type

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.

Properties: 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
```

For more information, see [“How to Create and Deploy a Non-Global Zone”](#) in *Creating and Using Oracle Solaris Zones*.

capped-cpu Resource Type

The `capped-cpu` resource type provides an absolute fine-grained limit on the amount of CPU resources that can be consumed by a project or a zone.

Properties: ncpus

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

When used in conjunction with processor sets, CPU caps limit CPU usage within a set. This resource sets a limit on the amount of CPU resources that can be consumed by the zone while it is running.

Note - When setting the `capped-cpu` resource type, 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.

ncpus Property

Specifies the limit on CPU usage as a positive decimal with two digits to the right of the decimal, which corresponds to units of CPUs. 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. The property does not accept a range.

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 type.

The following example specifies a CPU cap of 3.5 CPUs for the zone.

```
zonecfg:my-zone> add capped-cpu
zonecfg:my-zone:capped-cpu> set ncpus=3.5
zonecfg:my-zone:capped-cpu> end
```

capped-memory Resource Type

This resource type groups the properties used when capping memory for the zone. To use the capped-memory resource type, the service/resource-cap package must be installed in the global zone.

Properties:

locked	pagesize-policy (solaris-kz only)	physical	swap
--------	--------------------------------------	----------	------

- For solaris zones only, the capped-memory resource type provides limits for physical, swap, and locked memory. At least one of these properties must be specified.

The following example specifies the memory limits for the zone. Each limit is optional, but at least one limit 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
```

- For solaris-kz zones only, the capped-memory resource type provides limits for physical memory. The pagesize-policy specifies the policy for using large pages for physical memory.

The values for the `pagesize-policy` property are as follows:

`largest-available`

Provide the largest possible page size, scaling down the page size if the system cannot allocate all physical memory with a particular page size. The priority is to boot the zone.

`largest-only`

Allocate only the largest possible page size for the kernel zone's physical memory. If all the pages are not assigned, the zone cannot be booted.

`smallest-only`

Choose the lowest allowable page size required to boot the kernel zone for the particular platform.

The following example specifies the memory limits for the zone by using the `physical` and `pagesize-policy` properties.

```
zonecfg:my-kz> add capped-memory
zonecfg:my-kz:capped-memory> set physical=4g
zonecfg:my-kz:capped-memory> set pagesize-policy=largest-available
zonecfg:my-kz:capped-memory> end
```

If a kernel zone `pagesize-policy` property is cleared or not set, the zone uses the lowest allowable page size required to boot on the particular platform on which it is running. You must clear `pagesize-policy` if you want to migrate a kernel zone that was created in Oracle Solaris 11.4 to an earlier Oracle Solaris release that does not support the `pagesize-policy` property, such as the initial release of Oracle Solaris 11.3.

dataset Resource Type (solaris and solaris10 Only)

The only dataset type you can use 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.

Properties: `name`, `alias`

Note - Use the device resource type instead of the dataset resource type in kernel zones.

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.

The zone administrator can perform the following storage administration tasks:

- Create and destroy file systems within that dataset
- Modify properties of the dataset
- 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.

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 type is not specified, the default `zonpath` 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/`.

For more information, see [Chapter 10, “Oracle Solaris ZFS Advanced Topics” in *Managing ZFS File Systems in Oracle Solaris 11.4*](#), [“File Systems and Non-Global Zones” in *Creating and Using Oracle Solaris Zones*](#) and the `datasets(7)` man page.

For information about dataset issues, see [Chapter 11, “Troubleshooting Miscellaneous Oracle Solaris Zones Problems” in *Creating and Using Oracle Solaris Zones*](#).

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 the `clear alias` command.

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

dedicated-cpu Resource Type

This resource type dedicates a subset of the system's processors to the 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.

Note - The `dedicated-cpu` resource type is incompatible with the `capped-cpu` resource type and with the `cpu-shares` resource control.

Properties:

cores cpus importance ncpus sockets

The `dedicated-cpu` resource type provides limits for `ncpus` and, optionally, `importance`. It also sets persistent `dedicated-cpu` resources for `cores`, `cpus`, and `sockets`.

cores Property

Assign specific cores to zone persistently.

cpus Property

Assign specific CPUs to a zone persistently.

importance Property

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 the `poold` service is not running, then the `importance` property 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 11.4*](#).

ncpus Property

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 [“Enabling and Disabling the Pools Facility”](#) in *Administering Resource Management in Oracle Solaris 11.4*.

sockets Property

Assign specified number of sockets persistently.

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

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 type.

The following example specifies a CPU range for use by the zone and sets the `importance` level.

```
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
```

The following output shows the granularity of assigning core, cpu, socket, and a locality groups to a zone:

```
global$ psrinfo -tL
socket: 0 (lgroups: 1, 0)
  core: 0
    cpus: 0-7
  core: 1
    cpus: 8-15
  core: 2
    cpus: 16-23
  core: 3
    cpus: 24-31
  ...
  core: 7
    cpus: 56-63
```

```

socket: 1 (lgroups: 2, 0)
  core: 8
  cpus: 64-71
...
  core: 15
  cpus: 120-127

```

The following example persistently assigns cores 0, 1, 2, and 3 as a range to the zone. The example could potentially also have set values for `cpus=`, `sockets=`, and locality groups. For more information about locality groups, see [“Adding CPUs from a Locality Group to a Kernel Zone” in *Creating and Using Oracle Solaris Kernel Zones*](#).

```

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

```

device Resource Type

The device resource type is used to add virtual disks to a non-global zone's platform. The device resource type 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 type, the `system/file-system/ufs` package must be installed into the zone after installation or through the AI manifest script.

Properties:

<code>allow-mhd</code>	<code>allow-raw-io</code>	<code>storage</code>
<code>allow-partition</code>	<code>match (solaris only)</code>	

The `match` property is only supported on `solaris` zones. For `solaris-kz` zones, use the `storage` property.

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:

<code>%{zonename}</code>	<code>%{id}</code>	<code>%{global-rootzpool}</code>
--------------------------	--------------------	----------------------------------

For kernel zones, the following properties can be set to `true` or `false`; the default is `false`:

- The `allow-partition` property enables partitioning.

- The `allow-raw-io` property enables `uscsi`.
- The `allow-mhd` property enables multi-host disk control operations.

For more information about these resources, see the [solaris\(7\)](#), [solaris-kz\(7\)](#), and [zonecfg\(8\)](#) man pages.

Note - Do not use the `match` resource property for `solaris-kz` zones. Use the `storage` property. The `storage` property and storage URI are used to specify disk LUNs for kernel zones. In existing zone configurations, The `match` property is converted to `storage` automatically.

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 the `add device` command.

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:///lunname.naa.600144f03d70c80000004ea57da10001
zonecfg:my-zone:device> set bootpri=0
zonecfg:my-zone:device> end
```

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

```
device 0:
  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, review “[Device Use 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 Resource Type

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 type 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 type, 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 the [`quota\(8\)`](#) man page cannot be used to retrieve quota information for UFS file systems added through the `fs` resource type.

Properties:

`dir` `options` `raw` `special` `type`

The `fs` resource type properties supply the values that determine how and where to mount file systems. The `fs` properties are defined as follows:

`dir` Property

Specifies the mount point for the file system

`options` Property

Specifies mount options similar to those found with the `mount` command

`raw` Property

Specifies the raw device on which to run `fsck` before mounting the file system (not applicable to ZFS)

`special` Property

Specifies the block special device name or directory from the global zone to mount

`type` Property

Specifies the file system type

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 used 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 about parameters, see the following documentation:

- [“Security Restrictions and File System Behavior”](#) in *Creating and Using Oracle Solaris Zones*
- [mount\(8\)](#) man page
- [fsck\(8\)](#) man page
- Section 8 man pages for mount options of a specific file system. The names of these man pages have the form `mount_<filesystem>(8)`, for example, [mount_nfs\(8\)](#).

Note - The `quota` command documented in the [quota\(8\)](#) man page cannot be used to retrieve quota information for UFS file systems added through this resource type.

ib-vhca Resource Type (solaris-kz Only)

The `ib-vhca` resource type 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 11.4](#).

Properties:

<code>id</code>	<code>over-hca</code>	<code>port</code>	<code>smi-enabled</code>
-----------------	-----------------------	-------------------	--------------------------

The `ib-vhca` resource type specifies the physical function (PF) that is used to allocate a virtual function (VF). The `ib-vhca` resource type is supported in Live Zone Reconfiguration of a kernel zone.

`id` Property

Unique identifier for the `ib-vhca` resource type.

`over-hca` Property

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

`port` Property

Use the `port` property to specify the allowable pkeys for the allocated VF.

id Value

The `id` value is used to uniquely identify the port resource. The `id` corresponds to the physical port number, which is typically 1 or 2.

pkey Value

Specifies the InfiniBand Partition key value. The `pkey` value can be either of the following:

- The keyword `auto` - Use the `auto` keyword to automatically generate and assign a `pkey` value based on the `over-hca` value specified.
- A comma-separated list of hexadecimal values - Do not use the `0x` prefix to specify the hexadecimal value.

smi-enabled Property

Specifies whether the virtual HCA can use Subnet Management Packets (SMPs). The default value, which does not have to be explicitly set, is `off`.

- If the value of this property is `on`, SMPs are allowed for this virtual HCA.

Note - Do not use the value `on` without considering the possible security impact on the fabric. If `smi-enabled` is `on`, a kernel zones user can do SMP set operations through the `ib-vhca` resource type. A physical port can be disabled by SMPs that set the port state.

When using the `on` setting, `M_Keys` should be set on fabric components.

- If the value is `off`, SMPs cannot be used with this virtual HCA.
- If the value is `readonly`, this virtual HCA can use query SMP operations, but not set operations.

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

1. Virtualize the PF by using the `ibadm` command as described in the [ibadm\(8\)](#) 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 the `zoneadm` command. If a VF is not available, the resource allocation fails.

The following example adds an `ib-vhca` resource type to the running kernel zone for a device named `dev0`. This virtual HCA can only query Subnet Management Packets (SMP) operations. This virtual HCA cannot set operations.

```
global$ pfbash zonecfg -z zone-kz
zonecfg:zone-kz> add ib-vhca
zonecfg:zone-kz:ib-vhca> set over-hca=dev0
```

```
zonecfg:zone-kz:ib-vhca> set smi-enabled=readonly
zonecfg:zone-kz:ib-vhca> end
zonecfg:zone-kz>end
global$ zonecfg -z zone-kz apply
```

keysource Resource Type

Encryption key. Provides administrative access to the cryptographic key used for kernel zone suspend images and host data as described in the [solaris-kz\(7\)](#) man page.

Properties: raw

The value of the raw property cannot be set directly, except with the `command_file` mode.

net Resource Type

The net resource type assigns an existing network interface in the global zone to the non-global zone. The network interface resource type 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.

Properties:

address	configure-allowed-address	id
allowed-address	defrouter	physical

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.

allowed-address Property

The allowed-address property limits the set of configurable IP addresses that can be used by an exclusive-IP zone.

defrouter Property

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 datalinks 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 - Oracle Solaris supports all Ethernet-type interfaces. You can administer datalinks with the `dladm` command.

npiv Resource Type

Provides `N_Port_ID` Virtualization (NPIV) support in Oracle Solaris zones and Oracle Solaris 10 Zones.

Properties: `over-hba`, `virtual-port-wwn`

The `npiv` resource type is used to configure zones that have NPIV fibre channel devices as back-end storage for the zone root file system, and use other devices for data.

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` and `over-hba` properties are both optional.

The `virtual-port-wwn` property 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.

The following example delegates two `npiv` resources to the zone. 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
```

rctl Resource Type

The `rctl` resource type 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 33](#) for more information.

Note - To configure zone-wide controls using the `set global-property-name` subcommand of `zonecfg` instead of using the `rctl` resource type, see [“How to Create and Deploy a Non-Global Zone” in *Creating and Using Oracle Solaris Zones*](#).

Properties: `name`, `value`

The following zone-wide resource controls are available.

```

zone.cpu-cap resource control
zone.cpu-shares resource control (preferred: cpu-shares global property)
zone.max-adi-metadata-memory resource control
zone.max-locked-memory resource control
zone.max-lofi resource control
zone.max-lwps resource control (preferred: max-lwps global property)
zone.max-msg-ids resource control (preferred: max-msg-ids global property)
zone.max-processes resource control (preferred: max-processes global property)
zone.max-sem-ids resource control (preferred: max-sem-ids global property)
zone.max-shm-ids resource control (preferred: max-shm-ids global property)
zone.max-shm-memory resource control (preferred: max-shm-memory global property)
zone.max-swap resource control

```

The simplest method to set a zone-wide resource control is to use the global property name instead of the `rctl` resource control name, as shown in [“How to Create and Deploy a Non-Global Zone”](#) in *Creating and Using Oracle Solaris Zones*. If zone-wide resource control entries in a zone are configured by using the `zonecfg add rctl` command, the format is different than resource control entries made in the project database. In a zone configuration, an `rctl` resource consists of three name-value pairs. The names are `priv`, `limit`, and `action`. Each name takes a simple value. For example:

```

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

```

For general information about resource controls and attributes, see [Chapter 6, “About Resource Controls”](#) in *Administering Resource Management in Oracle Solaris 11.4* and [“Resource Controls Used in Non-Global Zones”](#) in *Creating and Using Oracle Solaris Zones*.

rootzpool Resource Type (solaris and solaris10 Only)

Identify the storage object URI to provide a dedicated ZFS storage pool (zpool) for zone installation.

Properties: storage

The optional `rootzpool` resource type is used to create a dedicated ZFS storage pool for zone installation for `solaris` 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` resource can be defined for a given zone. The storage is automatically configured for the zone when the zone is booted.

The corresponding `zpool`s are automatically created or imported during zone installation or zone attach operations. To reuse a pre-created `zpool` for a zone installation, the `zpool` must be exported from the system. The name `my-zone_rpool` is assigned. For both the `rootzpool` and `zpool` resource types, you can automatically create `zpool` mirrors as soon as the zone is installed. For more information, see [Chapter 12, “Oracle Solaris Zones on Shared Storage”](#) in *Creating and Using Oracle Solaris Zones*.

storage Property

The storage property supports the following URI types:

dev

Local device path URI. Formats are as follows:

```
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). Formats are as follows:

```
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. Formats are as follows:

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

Examples:

```
iscsi:///lunname.eui.0021280001cf80f6
iscsi:///lunname.naa.600144f03d70c80000004ea57da10001
iscsi://[:1]/lunname.naa.600144f03d70c80000004ea57da10001
iscsi://127.0.0.1/lunname.naa.600144f03d70c80000004ea57da10001
iscsi://hostname:1234/lunname.eui.0021280001cf80f6
iscsi://hostname:3260/lunname.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 following configuration example adds a `rootzpool` resource with two storage devices, for creating a mirrored configuration:

```
zonecfg:my-zone> add rootzpool
zonecfg:my-zone:rootzpool> add storage dev:dsk/c4t1d0
zonecfg:my-zone:rootzpool> add storage dev:dsk/c4t3d0
zonecfg:my-zone:rootzpool> end
```

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 `suriadm` tool is used to administer shared objects that are based on storage URIs. For information about IDs, the Name Address Authority (NAA), and obtaining URIs for existing storage objects, see the [suriadm\(8\)](#) and [suri\(7\)](#) man pages.

smf-dependency Resource Type

The set of Oracle Solaris Service Management Facility (SMF) Fault Management Resource Identifiers (FMRI) that establish the dependencies for the zone are defined in the smf-dependency resource type.

Properties: fmri, name, grouping

```
global$ pfbash zonecfg -z my-zone
zonecfg:my-zone> add smf-dependency
zonecfg:my-zone> set name=firewall
zonecfg:my-zone> set fmri=svc:/system/zones/zone:appfirewall
zonecfg:my-zone> set fmri=svc:/system/zones/zone:appnat
zonecfg:my-zone> set grouping=require_any
zonecfg:my-zone> end
zonecfg:my-zone> exit
```

All SMF dependencies for a zone will be of type service and have restart_on set to none.

For more information, see [“Zones and the Service Management Facility”](#) in *Creating and Using Oracle Solaris Zones* and the `zonecfg(8)` man page.

verified-boot Resource Type

The verified-boot resource type controls a kernel zone's boot policy and certificate settings. The properties are:

cert Property

The cert property specifies the location of the [elfsign\(1\)](#) X.509 public key certificate on the system. You specify the certificate location with a URI of the X.509 cert file. For a local file, the certificate must be located in the global zone's file system. For remote URIs, the URI must be accessible from the global zone.

Use the add subcommand to add a certificate. You can add up to seven certificates on each kernel zone.

policy Property

The policy property regulates the verification of the unix, genunix and other kernel modules. The possible values for this property are as follows:

enforce

Prints a warning message if `elfsign` signature verification fails. The kernel module does not load.

none

No action occurs if `elfsign` signature verification fails.

Note: A `verified-boot` resource is not enabled if the `policy` value is set to `none`.

warning

Prints a warning message if `elfsign` verification fails. This is the default value.

EXAMPLE 7 Enabling Verified Boot in a Kernel Zone

This example creates the kernel zone `kz1` on the system `global`. The `verified-boot` policy value is set to `enforce`. This directs the kernel to not boot if boot file signature verification fails and to print an error message on failure.

```
global$ pfbash zonecfg -z kz1
kz1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:kz1> create -t SYSsolaris-kz
zonecfg:kz1> set zonepath=/rpool/zones/kz1
zonecfg:kz1> set autoboot=true
zonecfg:kz1> add verified-boot
zonecfg:kz1:verified-boot> set policy=enforce
zonecfg:kz1:verified-boot> end
zonecfg:kz1> verify
zonecfg:kz1> commit
zonecfg:kz1> exit
```

EXAMPLE 8 Configuring Kernel Zone Verified Boot With Multiple Certificates

This example demonstrates adding a `verified-boot` resource to an already-configured kernel zone `kz2` on the host system `global`. Two certificates are added to the configuration.

```
global$ pfbash zonecfg -z kz2
zonecfg:kz2> add verified-boot
zonecfg:kz2:verified-boot> set policy=warning
zonecfg:kz2:verified-boot> add cert file:///etc/certs/elfsign/SOLARIS-KZ
zonecfg:kz2:verified-boot> add cert http://example/keydist/cert.pem
zonecfg:kz2:verified-boot> info
verified-boot:
  policy: warning
```

```
cert: file:///etc/certs/elfsign/SOLARIS-KZ
cert: http://example/keydist/cert.pem
zonecfg: kz2:verified-boot> end
zonecfg: kz2> verify
zonecfg: kz2> commit
```

virtual-cpu Resource Type (solaris-kz Only)

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 resource dedicates a subset of the system's processors to the zone while it is running. The `virtual-cpu` resource provides limits for `ncpus`.

Properties: `ncpus`

The default kernel zone configuration has 4 VCPUs. Each `virtual-cpu` can use up to 1 host CPU of compute power, but could get less if there is contention for host CPU resources. The host 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 either the `dedicated-cpu` resource type or the `pool` global property, then the number of virtual CPUs created match the size of that pool. 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 host system "de-schedules" all or part of the kernel zone.

Stolen time indicates the time when the kernel zone cannot run because the host 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 hosts 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 the `zonestat`, `mpstat`, `iostat`, and `vmstat` commands and by other utilities.

`ncpus` Property

Specifies the number of CPUs.

Note - 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.

The following example specifies 3 CPUs for the zone.

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

zpool Resource Type (solaris and solaris10 Only)

Define one or more storage object URIs to delegate a zpool to the zone.

Properties: name, storage

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

name Property

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_zpool-name* when displayed from the global zone.

The allowed values for the name property are defined in the [zpool\(8\)](#) man page.

storage Property

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*

Note - A zone installation can fail when a storage object contains preexisting partitions, zpools, or UFS file systems. For more information, see [“Troubleshooting Installation” in *Creating and Using Oracle Solaris Zones*](#).

For information about URIs and the allowed values for the storage property, see [“rootzpool Resource Type \(solaris and solaris10 Only\)” on page 69](#).

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.

In the following 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 `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 ; end
```

◆◆◆ APPENDIX A

Resource Types and Global Properties That Support Live Zone Reconfiguration

Live Zone Reconfiguration Support or Restriction

The following table compares Live Zone Reconfiguration support for resource types and global properties in non-global zones and in kernel zones.

Changes can be made either on a temporary or on a persistent basis. Temporary changes are active until the next reboot. You do not need to reboot for changes to be applied to the persistent configuration.

TABLE 3 Live Zone Reconfiguration Support or Restriction for Resource Types and Global Properties

Resource Type or Global Property	Native (solaris) Zone Support	Kernel Zone Support
admin Resource Type	Persistent only	Persistent only
anet Resource Type	Yes, with exceptions. No support for the following: anet:allowed-addresses anet:configure-allowed-addresses anet:defrouter anet:evs anet:vport	Yes, with exceptions. No support for the following: anet:allowed-addresses anet:configure-allowed-addresses anet:defrouter anet:evs anet:vport
attr Resource Type	Persistent only	Persistent only
autoboot Global Property	Persistent only	Persistent only
autosutdown Global Property	Persistent only	Persistent only
brand	No	No
bootargs Global Property	Persistent only	Persistent only
capped-cpu Resource Type	Yes	No
capped-memory Resource Type	Yes	No

Resource Type or Global Property	Native (solaris) Zone Support	Kernel Zone Support
cpu-arch Global Property	No	No
cpu-shares	Yes	No
dataset Resource Type	Yes	N/A
dedicated-cpu Resource Type	Yes	No
device	Yes	Yes
description Global Property	Persistent only	Persistent only
file-mac-profile Global Property	No	No
fs Resource Type	Yes	N/A
fs-allowed Global Property	No	No
global-time Global Property	No	No
ib-vhca Resource Type	No	Yes, including ib-vhca:port
limitpriv Global Property	No	No
net Resource Type	Yes with exceptions No support for the following: net:allowed-addresses net:configure-allowed-addresses net:defrouter	Yes with exceptions No support for the following: net:allowed-addresses net:configure-allowed-addresses net:defrouter
npiv Resource Type	No	No
pool Global Property	Yes	No
scheduling-class Global Property	Yes	No
tenant	No	No
virtual-cpu Resource Type	N/A	Yes
zonename Global Property	No	No
zonpath Global Property	No	No
zone-wide resource controls	Yes	No
zpool Resource Type	No	No

For more information about using Live Zone Reconfiguration, review the following:

- [Unresolved link to " Live Zone Reconfiguration of Kernel Zones in Creating and Using Oracle Solaris Kernel Zones"](#)
- [solaris\(7\)](#) man page
- [solaris-kz\(7\)](#) man page

Index

A

- ADI
 - enabling, 42
- allowed-addresses
 - exclusive-IP zone, 23
- anet resource type, 48
- attr resource type
 - adding a comment, 55
- autoboot global property, 36
- autoshutdown global property, 36

B

- boot-disk-protection global property, 37
- boot-priority global property, 37
- bootargs global property, 36
- brand global property, 36

C

- capped-cpu resource type, 55
- capped-memory resource type, 16, 56
 - pagesize-policy, 56
- comments
 - adding in a zone, 55
- configurable privileges, zone, 32
- configuring with zone, 54
- cpu-arch global property, 37
- cpu-shares global property, 39
- cross-CPU migration
 - kernel zones, 18, 19

D

- datalinks, 22
- dataset resource type, 57
- DAX
 - kernel zones, 42
- dedicated-cpu resource type, 59
- defrouter, 66
 - exclusive-IP zone, 23
- description global property, 39
- device resource type, 61
- Device Resources
 - with storage URIs, 30
- DHCP
 - exclusive-IP zone, 23
- disk format support
 - zones, 30
- dttrace_proc privilege, 43
- dttrace_user privilege, 43

E

- enabling
 - ADI, 42
 - Silicon Secured Memory (SSM), 42
- EVS with zones, 22
- exclusive-IP zone, 23

F

- fair share scheduler (FSS), 16
- features
 - exclusive-IP zone, 23
- fs resource type, 63

G

- global properties
 - autoboot, 36
 - autoshtutdown, 36
 - boot-disk-protection, 37
 - boot-priority, 37
 - bootargs, 36
 - brand, 36
 - configurable, 15
 - cpu-arch, 37
 - cpu-shares, 39
 - cross-CPU migrations, 19
 - description, 39
 - fair share scheduler (FSS), 16
 - file-mac-profile, 39
 - fs-allowed, 40
 - global-time, 40
 - host-compatible, 20, 41
 - hostid, 41
 - hwprovider, 43
 - ip-type, 43
 - limitpriv, 43
 - max-adi-metadata-memory, 44
 - max-lofi, 44
 - max-lwps, 44
 - max-processes, 44
 - max-System-V-resource, 44
 - pool, 45
 - scheduling-class, 45
 - supported tokens, 14
 - tenant, 45
 - zone, 35
 - zonename, 46
 - zonename, 46
- global-time global property, 40

H

- host-compatibility levels
 - description, 20
 - migration and, 18

- host-compatible global property
 - kernel zones, 18, 20, 42
 - migration feature, 41
- hostid global property, 41
- hwprovider global property, 43

I

- ib-vhca resource type, 64
- IP routing
 - exclusive-IP zone, 23
- ip-type global property, 43
- IPMP
 - exclusive-IP zone, 23
- IPoIB, 54

K

- kernel zones
 - boot-disk-protection global property, 37
 - cross-CPU migration, 18
 - host-compatible global property, 18, 20, 20, 42
 - managing non-global zones in, 54
 - SPARC cross-CPU migration, 19
 - x86 cross-CPU migration, 20, 37
- keysource resource type, 66

L

- limitpriv global property, 43
- link-mode, 52
- Live Reconfiguration
 - supported resource types, non-global and kernel zones, 77
- locked memory cap, 17
- lofi device
 - removable, 29

M

- max-adi-metadata-memory global property, 44
- max-lofi global property, 44

max-lwps global property, 44
max-msg-ids global property, 44
max-processes global property, 44
max-semids global property, 44
max-shm-ids global property, 44
max-shm-memory global property, 44

N

net resource
 shared-IP zone, 25
net resource type, 66
 exclusive-IP zone, 23
new features
 hwprovider global property, 43
npiv resource type
 solaris brand, 68
 virtualization support, 67

P

Packet Filter
 exclusive-IP zone, 23
pagesize-policy, 18
physical memory cap, 17
pkey, 53, 54
pool global property, 45

R

rctl resource type, 68
read-only zone
 file-mac-profile global property, 39
read-only zone root, 39
Reliable Datagram Sockets (RDS), 26
removable lofi device, 29
resource controls
 zone-wide, 33
resource types
 capped-memory, 16
 configurable, 15
 supported tokens, 14

rights profiles, 47
rootzpool resource type
 solaris brand, 69

S

scheduling-class global property, 45
shared-IP zone, 25
Silicon Secured Memory (SSM)
 enabling, 42
smf-dependency resource type, 72
SSM *See* Silicon Secured Memory (SSM)
swap space cap, 17
sys_time privilege, 41

T

temporary pool, 59
tenant global property, 45
tokens
 supported by properties, 14

V

verified-boot resource type, 72
virtual-cpu resource
 kernel zones, 74
virtual-cpu resource type, 74
VLAN, 54

X

x86 cross-CPU migration
 kernel zones, 20, 37

Z

ZFS
 dataset resource type, 57
 fs-allowed global property, 40
zone
 anet resource type, 48

- attr resource type, 55
- autoshutdown global property, 36
- bootargs global property, 36
- brand global property, 36
- capped-cpu resource type, 55
- capped-memory resource type, 16, 56
- configurable privileges, 32
- configuration overview, 11
- configuring, 11
- cpu-shares global property, 39
- dataset resource type, 57
- dedicated-cpu resource type, 59
- description global property, 39
- device resource type, 61
- disk format support, 30
- exclusive-IP, 23
- fair share scheduler (FSS), 16
- fs resource type, 63
- fs-allowed global property, 40
- global properties, 35
- ib-vhca resource type, 64
- ip-type global property, 43
- keysource resource type, 66
- limitpriv global property, 43
- max-adi-metadata-memory global property, 44
- max-lofi global proper, 44
- max-lwps global proper, 44
- max-processes global proper, 44
- net resource type, 66
- npiv resource type, 67
- pool global property, 45
- rctl resource type, 68
- resource controls, 33
- resource type properties, 46
- resource types, 46
- rootzpool resource type, 69, 69
- scheduling-class global property, 45
- shared-IP, 25
- smf-dependency resource type, 72
- tenant global property, 45
- tokens, 13
- verified-boot resource type, 72
- virtual-cpu resource type, 74
- zone-wide resource controls, 35
- zonename global property, 46
- zonename global property, 46
- zpool resource type, 75
- zone admin authorization, 47
- zone rights profiles, 47
- zone-wide resource controls, 33
- zone.cpu-cap resource control, 33
- zone.cpu-shares resource control, 33
- zone.max-adi-metadata-memory resource control, 33
- zone.max-locked-memory resource control, 33
- zone.max-lofi resource control, 34
- zone.max-lwps resource control, 34
- zone.max-msg-ids resource control, 34
- zone.max-processes resource control, 34
- zone.max-sem-ids resource control, 34
- zone.max-shm-ids resource control, 34
- zone.max-shm-memory resource control, 34
- zone.max-swap resource control, 34
- zonecfg
 - admin authorization, 47
 - autoboot global property, 36
 - boot-priority global property, 37
 - global-time global property, 40
 - in global zone, 11
 - operations, 11
 - pool global property, 45
 - properties and resource types, 35
 - template property, 13
- zonename global property, 46
- zonename global property, 46
- zpool resource type, 75