

Creating and Using Oracle Solaris Zones



E61039-01
November 2021



Creating and Using Oracle Solaris Zones,

E61039-01

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

Primary Author: Cathleen Reiher, Sharon Veach

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, software documentation, data (as defined in the Federal Acquisition Regulation), 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," "commercial computer software documentation," or "limited rights data" 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®, Java, and MySQL 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.

Contents

Using This Documentation

Product Documentation Library	xi
Feedback	xi

1 Before You Begin Working With Oracle Solaris Zones

Planning a Non-Global Zone Configuration	1-1
Hardware Requirements	1-1
Zone Size	1-1
Name Service and Network Considerations	1-2
About Zone Brands	1-2
Zone Installation and Configuration Concepts	1-3
About Zone Properties	1-3
About Zone Commands for Configuration and Installation	1-3
Using Manifests and Configuration Profiles	1-4
Creating a Manifest for Zone Installation	1-4
Creating a System Configuration Profile for Zone Installation	1-4
Using Rights Profiles to Install and Manage Zones	1-5

2 Setting Up a Non-Global Zone

Installing a Non-Global Zone	2-1
How to Create and Deploy a Non-Global Zone	2-1
Displaying Zone-Related Information	2-4
Using the zoneadm Command	2-5
Using the zonecfg Command	2-5
Using the zlogin Command	2-6
Installing Non-Global Zones Based on Existing Zones	2-7
Using Clones	2-7
Using Unified Archives	2-7
How to Install and Deploy Zones by Cloning	2-8
Uninstalling and Deleting a Non-Global Zone	2-10

3 Managing Zone Resources and Properties

About Configuring Zone Properties	3-1
How to Configure Zone Properties and Resources	3-2
Setting Values of Zone Properties and Resources	3-3
Limiting the Zone's Resources	3-3
Assigning CPUs to the Zone	3-3
Setting Cap Limits for Memory	3-4
Adding File Systems for Zone Use	3-4
Setting the hwprovider Property to Override the Global Zone Value (solaris10 Only)	3-6
Setting Time Values in Non-Global Zones	3-6
Modifying Zone Property and Resource Settings	3-7
Canceling a Configuration Session	3-7
How to Modify Zone Resources and Properties	3-7
Configuring the dataset Property on solaris Zones	3-11
Performing Zone Configuration From the Global Zone	3-13

4 Administering Networking in Oracle Solaris Non-Global Zones

About Networking In Zones	4-1
How to Create a Private Network of Multiple Zones	4-1
Using IP Network Multipathing With Non-Global Zones	4-3
How to Extend IP Network Multipathing Functionality to Shared-IP Non-Global Zones	4-4
Managing Datalinks Related to Zones	4-5
Changing Existing Shared-IP Zones to Exclusive-IP Zones	4-5

5 Administering SMF and Reporting in Oracle Solaris Non-Global Zones

Zones and the Service Management Facility	5-1
Zone SMF Dependencies	5-1
About the Zone SMF Instance Service State	5-2
Zone SMF Instance Service States	5-3
SMF Goal Services and Zones	5-4
Zones Boot Ordering	5-4
Administering the Zones Delegated Restarter	5-5
Zones Delegated Restarter	5-5
How the Zones Delegated Restarter Works	5-5
Zones Delegated Restarter Properties	5-6
Reporting the Status of SMF Services in a Non-Global Zone	5-6
Reporting Resource Usage in a Non-Global Zone	5-7

Reporting Per-Zone File System Statistics	5-9
Using the Fair Share Scheduler on a System With Zones Installed	5-10
How to Temporarily Set FSS Shares in the Global Zone	5-10

6 About Non-Global Zone Installation and Packages

Packages and Zones on an Oracle Solaris 11.4 System	6-1
Configuring Proxies to the Package Repository for Non-Global Zones	6-2
Overriding system-repository Proxies by Using https_proxy and http_proxy	6-2
Zone Contents After Creation	6-3
Zone Information After Installation	6-5
zoneadmd Zones Administration Daemon	6-5
zsched Zone Scheduler Process	6-5
Boot Environments and Zones	6-5
beadm Command in Non-Global Zones	6-5
Boot Environments in Global and Non-Global Zones	6-6
-x deny-zbe-clone Option	6-6
-x destroy-orphan-zbes Option	6-6
-x force-zbe-clone Option	6-7
Shared Zone State	6-7
Specifying an Alternate Root	6-7
About Orphaned Zone Boot Environments and Clones	6-7
Identifying Orphaned Boot Environments With the beadm Command	6-7
zoneadm Options for Orphaned Zone Boot Environments and Clones	6-8

7 About Non-Global Zone Administration

Security Measures for a System With Non-Global Zones	7-1
Privileges in a Non-Global Zone	7-2
Device Use in Non-Global Zones	7-3
/dev and the /devices Namespace	7-3
Exclusive-Use Devices and Zones	7-3
Device Driver Administration in Zones	7-4
Device Utilities and Non-Global Zones	7-4
Device Utilities That Do Not Work in Non-Global Zones	7-4
SPARC: eeprom in Non-Global Zones	7-4
Device Utilities With Security Implications	7-4
Resource Management With Non-Global Zones	7-5
Resource Controls Used in Non-Global Zones	7-5
Fair Share Scheduler on a System With Zones Installed	7-5
Share Balance Between Zones	7-5

Extended Accounting on a System With Zones Installed	7-6
Monitoring a System With Non-Global Zones	7-6
Global Zone Visibility and Access	7-6
Process ID Visibility in Zones	7-7
System Observability in Zones	7-7
Reporting Active Zone Statistics With the zonestat Utility	7-7
Monitoring Non-Global Zones With the fsstat Utility	7-8
Running DTrace in a Non-Global Zone	7-8
Core Files in Zones	7-8
Non-Global Zone Node Name	7-8
File Systems and Non-Global Zones	7-9
Running an NFS Server in a Zone	7-9
Mounting File Systems in Zones	7-9
Unmounting File Systems in Zones	7-10
Non-Global Zones as NFS Clients	7-11
Traversing File Systems	7-11
Security Restrictions and File System Behavior	7-11
Restriction on Using mknod in a Zone	7-13
Restriction on Accessing a Non-Global Zone From the Global Zone	7-13
Networking in Exclusive-IP Non-Global Zones	7-14
Overview of Networking in Exclusive-IP Zones	7-14
Exclusive-IP Zone Partitioning	7-14
Exclusive-IP Datalink Interfaces	7-14
Exclusive-IP Zones Traffic, Traffic Security, and IPMP Configuration	7-15
Networking in Shared-IP Non-Global Zones	7-15
Shared-IP Zone Partitioning	7-15
Shared-IP Network Interfaces	7-16
Shared-IP Zones Traffic, Traffic Security, and IPMP Configuration	7-16
Migration and Transformation of Non-Global Zones	7-17
About Cold Migration of solaris Zones	7-17
Zone Migration Configuration Requirements	7-17
zoneadm migrate Command	7-18
Zone Migration and Various Host Configurations	7-18
Zone Migration and Package Updates	7-18
About Zone Transformations	7-19
Zone Transformations Using Oracle Solaris Unified Archives	7-19
Zone Transformation Requirements for Source and Target Hosts	7-20
Zone Transformations From Systems That Contain Non-Global Zones	7-20
Using the zonep2vchk Tool to Prepare for Global to Non-Global Transformations	7-21
About the zonep2vchk Tool	7-21
Checks Performed by the zonep2vchk Tool	7-22

Issues Highlighted by the zonep2vchkTool	7-22
Zone Configuration Template for a Transformed Zone	7-23
Backup Strategies for a System With Non-Global Zones	7-23
Backup Methods for a System With Non-Global Zones	7-23
Backing Up Loopback File System Directories	7-23
Backing Up Your System From the Global Zone	7-23
Backing Up Individual Non-Global Zones on Your System	7-24
Creating Oracle Solaris Unified Archive Backups	7-24
Determining What to Back Up in Non-Global Zones	7-24
Backing Up Application Data Only	7-24
General Database Backup Operations	7-25
Tape Backups	7-25
About Restoring Non-Global Zones	7-25
Backing Up and Recreating Zones	7-26
Commands Used on a System With Non-Global Zones	7-26
Commands for Zone Administration and Monitoring	7-26
Commands for Resource Capping	7-27
Commands With Modifications for Zones	7-27

8 Migrating an Oracle Solaris Zone

Authorizing Non-Root Users to Perform Non-Global Zone Migrations	8-1
How to Authorize a User to Perform Cold Migration of an Individual Zone	8-1
Using Cold Migration to Migrate a solaris Zone	8-3
How to Cold Migrate a solaris Zone	8-3
Migrating a Zone From a System That Is Not Usable	8-4
Archiving and Moving Non-Global Zones That Are Not Using Shared Storage	8-5
How to Use Unified Archive to Move a Non-Global Zone to a New System	8-5
How to Move a Zone That Is Not on Shared Storage	8-6

9 Live Zone Reconfiguration of Oracle Solaris Zones

Live Zone Reconfiguration of Oracle Solaris Zones	9-1
zonecfg Utility Edit Modes	9-1
Temporary Changes to the Live Zone Configuration	9-2
Persistent Changes to the Live Zone Configuration	9-2
Live Zone Reconfiguration Dry Run	9-2
Reloading a Live Zone Configuration	9-3
Performing a Live Zone Reconfiguration	9-3
How to Inspect the Live Configuration of a Running Zone	9-3
How to Preview the Effect of a Live Zone Configuration	9-4

How to Make Persistent Configuration Changes to a Live Zone	9-4
How to Make Temporary Changes to the Running Zone	9-5
How to Recover From a Failure While Committing Live Zone Configuration Changes	9-6
Live Zone Reconfiguration of Datasets	9-7
How to Add a ZFS Dataset to a Running Zone	9-7
How to Remove a dataset Resource From a Running Zone	9-9
How to Export a Dataset From a solaris Zone	9-10

10 Transforming Systems to Oracle Solaris Zones

Transforming a Global Zone Into a solaris Non-Global Zone	10-1
How to Check the Source System With zonep2vchk	10-1
How to Create an Archive of the System Image	10-2
How to Configure the Transformed Zone on the Target System	10-3
How to Install the Transformed Zone From the Archive on the Target System	10-4

11 Configuring and Administering Immutable Zones

About Immutable Zones	11-1
Configuring Immutable Zones	11-1
Setting the MWAC Security Policy	11-2
Zone Resource Exceptions to MWAC Security Policy	11-3
SMF Services in Immutable Zones	11-3
Administering Immutable Non-Global Zones	11-5
Determining Whether a Non-Global Zone Is Immutable	11-5
Methods for Administering Non-Global Immutable Zones	11-5
Administering an Immutable Zone by Using the Trusted Path Domain	11-6
How to Enable Administrative Access to an Immutable Zone From the Console	11-6
How to Enable Remote Administrative Access to an Immutable Zone by Using RAD	11-7
Administering an Immutable Zone by Making It Writable	11-8
Immutable Global Zones	11-8
Configuring an Immutable Global Zone	11-9
Maintaining an Immutable Global Zone	11-9

12 Troubleshooting Miscellaneous Oracle Solaris Zones Problems

Installation Fails Due to Unmatched Allowable Packages	12-1
Troubleshooting Installation	12-1
Exclusive-IP Zone Is Using Device, so dladm reset-linkprop Fails	12-2
zonecfg -r Fails	12-2
Troubleshooting a Zone That Has Failed	12-3

About Zone UUIDs	12-3
How to Manage a Zone That Has Become Unavailable	12-3
Insufficient Privileges Specified for the Zone	12-5
Zone Does Not Halt	12-5
Zone Access Denied	12-6

13 Oracle Solaris Zones on Shared Storage

About Shared Storage Resources Using Storage URIs	13-1
Local Device URI	13-1
Logical Unit URI	13-2
iSCSI URI	13-3
Managing Storage URIs and Shared Storage Resources	13-3
Assigning Shared Storage Resources to Oracle Solaris Zones	13-5
storage Property for Zones	13-5
rootzpool Resource	13-6
zpool Resource	13-6
Renaming Zones	13-7
Relocating Shared Storage Resource	13-7
Automated ZFS Storage Pool Management for Oracle Solaris Zones on Shared Storage Resources	13-7
About the unavailable Zone State on Shared Storage	13-8
zoneadm Options for Shared Storage	13-8
Options for Installing, Cloning, and Attaching Zones	13-8
-x force-zpool-import Option	13-9
-x force-zpool-create= Option	13-9
-x force-zpool-create-all Option	13-9
Options for Uninstalling Zones	13-9
-x force-zpool-destroy=zpoolname Option	13-9
-x force-zpool-destroy-all Option	13-10
-x force-storage-destroy-all Option	13-10
Restrictions on Moving a Zone With the rootzpool Resource	13-10
Implementing Oracle Solaris Zones Hosted on Shared Storage Resources	13-10
zpool Configuration Considerations for Zones on Shared Storage	13-10
Sample Scenarios for Zones on Shared Storage	13-11
Migrating Oracle Solaris Zones Hosted on Shared Storage Resources	13-21
Moving Existing Zones To or From Shared Storage Zone Configurations	13-23
How to Move a Zone to a Shared Storage Configuration	13-23
How to Move a Zone From a Shared Storage Configuration	13-24
How to Move a Zone to Different Shared Storage	13-25
How to Add or Move a ZFS Storage Pool to an Installed Zone	13-26

Glossary

Index

Using This Documentation

- **Overview** – Describes how to set up and use the Oracle Solaris Zones feature and related resource management capabilities
- **Audience** – Technicians, system administrators, and authorized service providers
- **Required knowledge** – Experience with installing Oracle Solaris operating systems and familiarity with automated installs (AI) and the use of AI manifests and system configuration profiles. 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>.

1

Before You Begin Working With Oracle Solaris Zones

This chapter describes what you need to do before you can configure a zone on your system. It covers the following topics:

- [Planning a Non-Global Zone Configuration](#)
- [Zone Installation and Configuration Concepts](#)
- [Using Rights Profiles to Install and Manage Zones](#)

Planning a Non-Global Zone Configuration

As virtual instances of a physical system, zones use the hardware resources of their hosts. Therefore, the more resources on the host system, the greater the flexibility for an administrator to create and configure zones.

The following sections discuss important considerations when planning to create zones.

Hardware Requirements

Zones use the following hardware resources:

- Disk space

Disk space requirements for a zone depend on two factors:

- Packages you intend to install on the zone.
- Available disk space on the host after the installation of the standard Oracle Solaris OS packages.

Calculate at a minimum 1GB of disk space to allocate to each zone that you plan to install.

- Amount of memory

As an option, consider adding 40 MB of system memory for each zone to be configured. Disk space resources on the host is a factor in configuring swap space. If you configure sufficient swap space on the host system, then additional memory might not be necessary.

Zone Size

As an administrator, you can further control the size of a zone by setting the `quota` and `reservation` ZFS properties, which determine the maximum amount of disk space that a zone can consume. The properties are set on the zone's dataset that is located on the `zonepath` defined for that zone. For more information about these properties, see the [zfs\(8\)](#) man page and [Chapter 7, Managing Oracle Solaris ZFS File Systems in *Managing ZFS File Systems in Oracle Solaris 11.4*](#).

As an alternative, you can create ZFS volumes with fixed sizes and install zones in the volume's datasets. The volumes limit the sizes of the zones installed on the volumes. See [ZFS Volumes in *Managing ZFS File Systems in Oracle Solaris 11.4*](#).

Name Service and Network Considerations

Assign a zone name by which the zone can be identified from the global zone and to distinguish it from other zones on the system. Later, during the configuration of the zone's system information, you would be prompted for a computer name or a host name, similar to how physical systems have host names.

If you are not using DHCP, then you need to assign fixed IP addresses to the zones. Zones support both IPv4 and IPv6 addresses. Use IPv6 stateless address autoconfiguration to assign IPv6 addresses more easily.

By default, zones are automatically created as having an exclusive IP configuration, which means the zones have their own network layout different from and separate from the layout of the global zone. See [About Zone Network Interfaces in *Oracle Solaris Zones Configuration Resources*](#).

About Zone Brands

Zones in Oracle Solaris are branded. A zone is branded when you create it, as specified in the zone's *template*. After a zone is installed, you can no longer change its brand. Oracle Solaris 11.4 supports the following brands:

- `solaris` – Creates a zone that will contain the same software as the global zone. Unless you specify differently, all non-global zones automatically receive the `solaris` brand, which is based on the `SYSsolaris` template.

 **Note:**

This guide focuses on the `solaris` brand.

- `solaris-kz` – Creates a kernel zone. The brand is based on the `SYSsolaris-kz` or `SYSsolaris-kz-minimal` template. See [Creating and Using Oracle Solaris Kernel Zones](#).
- `solaris10` – Enables Solaris 10 binary applications to run unmodified on a host with the Oracle Solaris 11.4 kernel. The brand is based on the `SYSsolaris10` template. See [Creating and Using Oracle Solaris 10 Zones](#).
- `labeled` – Implements Trusted Extensions. See [Zone Creation in Trusted Extensions in *Trusted Extensions Configuration and Administration*](#) and [Installing and Enabling Trusted Extensions in *Trusted Extensions Configuration and Administration*](#).

For more details about brands, see the `brands(7)` man page.

**Note:**

The brand of zone you are creating might have additional requirements. For example, kernel zones have their own specific hardware and software needs, as described in [Software and Hardware Requirements for Oracle Solaris Kernel Zones in *Creating and Using Oracle Solaris Kernel Zones*](#).

Zone Installation and Configuration Concepts

This section provides an overview of the installation process to ease the task of creating and deploying non-global zones.

About Zone Properties

Multiple zone properties can be configured for a zone. Together, these properties characterize the zone as a virtual machine that is distinct from other zones on the host.

When you create a non-global zone without specifying additional options other than the zone name, the zone is created with the following set properties:

- The zone is a `solaris` branded zone.
- The `zonepath`, which identifies the zone's dataset, is set to `/system/zones/zonename`.
- The datalink name of an automatic network interface (`anet`) resource is set to `net0`
- The zone's IP type is exclusive.

Some zone properties are grouped into a resource. For example, aside from `linkname`, the `anet` resource also consists of properties such as `allowed-address`, `auto-mac-address`, `mtu`, and so on.

You can install the zone with just the default property settings. However, you can configure additional resources and properties to make the zone more efficient.

Setting zone properties can be performed before or after the zone installation. However, the typical approach is to set basic zone properties before installing the zone, then configure additional properties after installation is completed and the zone is booted and running.

You can set zone properties through a single command line. However, using the interactive method can prevent errors in property configuration. The rest of the documentation uses the interactive method to illustrate zone configuration procedures.

For a list of zone resources and properties, see the [`zonecfg\(8\)`](#) man page or [Oracle Solaris Zones Configuration Resources](#).

About Zone Commands for Configuration and Installation

You use two main commands for installing and configuring zones. Each command has its respective subcommands and options. You issue these commands from the global zone.

zonectfg Command

In zone deployment, the `zonectfg` command is used to create a zone and set its properties and resources. Through its multiple subcommands and options, you can also use this command for later administrative tasks such as modifying or removing property configuration as well as verifying the validity of zone settings. For details on the use of the command for zone property configuration, see [Managing Zone Resources and Properties](#)

zoneadm Command

This command is used to start the installation process as well as to boot the zone after the installation is completed. Use the command also to display information about zones on the system, their respective status, and so on.

Using Manifests and Configuration Profiles

You can use a manifest to install a non-global zone just as you can use one to install a global zone. To see how a manifest or a system configuration profile is used during a zone installation, see [How to Create and Deploy a Non-Global Zone](#).

Creating a Manifest for Zone Installation

During a zone installation, a default zone manifest is automatically used that installs the `solaris-small-server` package. If you want to install additional packages, then you would need to create a customized zone manifest to instruct the install program accordingly. For more information, see [Chapter 7, Installing and Configuring Zones in Automatically Installing Oracle Solaris 11.4 Systems](#).

**Note:**

The configurable parameters in a zone manifest are restricted compared to the manifest for an Oracle Solaris system installation.

To create a manifest, do the following:

1. On the host system, create a copy of the `/usr/share/auto_install/manifest/zone_default.xml` file.
2. Edit the copy by specifying packages to install in addition to the `solaris-small-server` package.

For guidance, refer to [Non-Global Zone AI Manifest in Automatically Installing Oracle Solaris 11.4 Systems](#) and [Working with Non-Global Zones in Updating Systems and Adding Software in Oracle Solaris 11.4](#).

Creating a System Configuration Profile for Zone Installation

Any Oracle Solaris installation, whether on a physical system or on a zone, involves system configuration to set the system's computer or host name, the time and date format to use, time zone, default language, and so on. You manually specify these parameters in the System Configuration Interaction (SCI) tool that opens at the end of the installation.

As an option, you can skip the manual step and automate the system configuration process by creating a system configuration profile. The install program applies the profile's contents as part of the installation process.

Essentially, with a configuration profile, you are performing the manual steps ahead of time so you would not need to perform them later.

To prepare a configuration file, follow these steps:

1. Run the SCI tool to create a profile.

```
global$ sysconfig create-profile -o directory
```

The directory serves as the storage location of the profile.

2. Go through each screen of the SCI tool and provide the information as prompted.

After you save and exit the tool, the settings are stored in a profile called

```
sc_profile.xml.
```

 **Note:**

All profiles you create with the SCI tool are named `sc_profile.xml`. Thus, as best practice, rename the profiles later for easier identification, especially if you create multiple profiles for different zones.

For more information about system configuration profiles, see [Chapter 3, Working With System Configuration Profiles in Customizing Automated Installations With Manifests and Profiles](#).

Using Rights Profiles to Install and Manage Zones

Oracle Solaris implements role-based access control (RBAC) to control system access. To perform specific tasks and run privileged commands on the system, a user must have the profiles that provide that user the proper authorization. RBAC also applies when working with zones. This section describes how to authorize sample user `jdoe` to work with zones.

Provided that you have the `solaris.delegate.*` authorization, you can assign one or all of the following zone-related profiles to `jdoe`.

- Zone Configuration enables `jdoe` to configure Oracle Solaris zones on a client system, including zones that are migrated into the system but remain unconfigured.
- Zone Management enables `jdoe` to manage existing zones, including running `zoneadm` commands as well as logging in to the zones.
- Zone Migration enables `jdoe` to perform live, warm, and cold zone migrations.
- Zone Cold Migration enables `jdoe` to perform cold and warm migrations, but not live migrations.
- Zone Security extends `jdoe`'s zone administrative privileges to include zones that are configured with Trusted Extensions. For example, the user with this profile can issue `txzonemgr` commands. For information about `txzonemgr`, see [Creating Labeled Zones in Trusted Extensions Configuration and Administration](#)

A zone administrator that has any of these profiles must be in a profile shell to execute privileged zone commands. The shell can be created by issuing the `pf shell` command, for

example, `pfbash`. Outside a profile shell, the administrator would need to specify `pfexec` when issuing a privileged command, for example, `pfexec zoneadm`.

As an alternative to directly assigning profiles, you can create a role that would contain a combination of required profiles to perform a range of tasks, and then assign the role to the designated zone administrator.

For example, you create a special role `zoneadmin` with authorization to perform a variety of zone related tasks. User `jdoo` then issues the `su` command to assume the `zoneadmin` role to manage the zones. All roles automatically get `pfbash` as the default shell.

For more information about rights profiles, see [Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*](#).

Later, a zone administrator can implement RBAC inside the running zone to manage users, rights, profiles, and roles to secure the zone. See examples in [Authorizing Non-Root Users to Perform Non-Global Zone Migrations](#).

2

Setting Up a Non-Global Zone

This chapter describes how to set up a zone, from creating the zone to activating it as a virtual machine. This chapter covers the following topics:

- [Installing a Non-Global Zone](#)
- [Displaying Zone-Related Information](#)
- [Uninstalling and Deleting a Non-Global Zone](#)
- [Installing Non-Global Zones Based on Existing Zones](#)

For background information, review [Planning a Non-Global Zone Configuration](#).

Installing a Non-Global Zone

The following basic steps create and deploy a zone:

1. Create the zone, with or without some properties.
2. Install and then boot the zone.
3. If necessary, log in to the zone to perform system configuration.

How to Create and Deploy a Non-Global Zone

If you plan to install other packages in addition to the default `solaris-small-server` package, make sure that you have properly prepared the manifest. Also, if you want to automate the system configuration of the zone, make sure that you have created and prepared the configuration profile. See [Using Manifests and Configuration Profiles](#).

Unless specified otherwise, all the procedures in this guide assume that you are not working directly on the server system that hosts the zones. Rather, you are remotely connected to the system through a secure shell session.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Create the zone.

Choose from one of the following options.

- Create the zone with no additional options.

```
global$ zonecfg -z zonename create [-t template]
```

Specify the `-t template` option if you want a different brand from `solaris`.

- Create the zone and configure some zone properties.

a. Switch to the interactive configuration method and create the zone.

```
global$ zonecfg -z zonename
Use 'create' to begin configuring a new zone.
zonecfg:zonename> create
```

```
create: Using system default template SYSdefault
zonename>
```

b. Configure zone properties.

As an example for this step, set the `autoboot`, `description`, and `physical` properties. `physical` is a property of the `net` resource. `description` is a global property.

```
zonecfg:zonename> set autoboot=true
zonecfg:zonename> set description="zone description"
zonecfg:zonename> add net
zonecfg:zonename:net> set physical=net1
zonecfg:zonename:net> end
zonecfg:zonename> exit
global$
```

 **Note:**

If you want a different brand from `solaris`, you must also type `set brand=brand` before exiting interactive configuration mode.

3. From the global zone, install the non-global zone.

```
global$ zoneadm -z zonename install \
[-m full-path-to-manifest.xml] [-c full-path-to-profile.xml]
```

Specify the `-m` option if you are using a manifest or the `-c` option if you are using a configuration profile.

4. bootingzoneszoneadm boot commandBoot the zone.

```
global$ zoneadm -z zonename boot [-- -m verbose]
```

Use the `-m verbose` option to obtain more information than the standard command output. For other options of the `boot` command, see the `zoneadm(8)` man page.

If you used a configuration profile during the installation, skip the next and final step.

5. If you did not use a configuration profile in Step 3, manually perform the zone's system configuration.

a. Open a separate terminal window.

b. Log in to the zone console.

```
global$ zlogin -C zonename
[Connected to zone zonename console]
```

c. Press Enter.

The SCI tool opens.

d. On each of the SCI tool's screens, provide the information as prompted.

For the computer name, you can optionally assign a different name than the zone name.

When you exit the SCI tool, the login prompt is displayed.

```
zone-hostname login:
```

e. Log in by using the credentials that you specified in the SCI tool.

After logging in, the prompt would be similar to the following:

```
username@zone-hostname:~$
```

f. Log out of the zone.

```
username@zone-hostname:~$ exit
zone-hostname login:
```

g. Disconnect the console by using the tilde-dot characters.

```
zone-hostname login: ~.
zone-hostname login: Connection to host-system closed.
```

Example 2-1 Creating a Non-Global Zone With a System Configuration Profile

This example shows how a zone named `zoneA` is created based on the following assumptions:

- The zone uses the default `solaris` brand.
- The properties `zonepath` and `autoboot` are set.
- The `physical` property of the `net` resource is set.
- For its root file system, `zoneA` will use `rpool/zones/`, whose mountpoint is `/zones`.
- A profile is used to automate the zone's system configuration.
- Only the default package is installed, and therefore, no special manifest is created.
- All necessary rights, both zone related and non-zone related, have been granted.

```
global$ pfbash
                                Create the profile. Provide information in the SCI tool.
global$ sysconfig create-profile -o /var/tmp
                                Create the zone and set zone properties.
global$ zonecfg -z zoneA
Use 'create' to begin configuring a new zone.
zonecfg:zoneA> create
create: Using system default template 'SYSdefault'
zonecfg:zoneA> set autoboot=true
zonecfg:zoneA> set zonepath=/zones/zoneA
zonecfg:zoneA> add net
zonecfg:zoneA:net> set physical=net1
zonecfg:zoneA:net> end
zonecfg:zoneA> exit
global$

Install the zone.
global$ zoneadm -z zoneA install -c /var/tmp/sc_profile.xml
                                Boot the zone.
global$ zoneadm -z zoneA boot
global$
```

Example 2-2 Creating a Non-Global Zone With Default Settings

This example shows how a zone named `zoneB` is created with default settings and based on the following assumptions:

- The zone uses the default `solaris` brand.
- Only the default package is installed, and therefore, no special manifest is created.
- All necessary rights, both zone related and non-zone related, have been granted.
- Because no configuration profile is used, system configuration of the zone is completed manually.

```
global$ pfbash
                                Create the zone.
global$ zonecfg -z zoneB create
global$

Install the zone.
global$ zoneadm -z zoneB install
                                Boot the zone.
global$ zoneadm -z zoneB boot
                                Connect to console then configure system.
global$ zlogin -C zoneB
[Connected to zone 'zoneB' console]
...
SC profile successfully generated as
/etc/svc/profile/incoming/config/sc_profile.xml

Exiting System Configuration Tool. Log is available at
logfilename
Booting to milestone "all".
Hostname: zone-hostname
zone-hostname console login:
username
Password:
password
username@zone-hostname:~$

Log out of zone.
username@zone-hostname:~$ exit
logout

zone-hostname console login: ~.
zone-hostname console login: Connection to global-zone is closed.
```

Displaying Zone-Related Information

From the global zone, you can use the `zoneadm`, `zonecfg`, or `zlogin` command to display information about the zones. Each command displays different data about the zones. You must be a zone administrator to run these commands. If necessary, switch to a profile shell.



Tip:

From any zone, use the `zonename` command to display the name of the zone.

Using the `zoneadm` Command

The `zoneadm list` subcommand provides you information about the status of zones on the system.

```
global$ zoneadm list -cv
ID NAME      STATUS  PATH                                BRAND  IP
0  global    running /                                     solaris shared
1  zoneA     running /export/zones/zoneA               solaris exclusive
2  zoneB     running /system/zones/zoneB              solaris exclusive
```

The output of the command indicates the following about a zone's status.

- **configured:** Zone has been created, but no further action has been done. The zone will not have an ID.
- **incomplete:** An error occurred during the installation process.
- **installed:** Zone has been installed but not booted. The zone will not have an ID.
- **running:** Zone is booted and operational. Only a running zone is assigned an ID.

A non-global zone's ID is not permanent. It changes every time the zone is rebooted. However, the global zone's ID is always zero.

Note also the different paths for `zoneA` and `zoneB` in the preceding example. If the `zonepath` property is not explicitly defined, as in the case of `zoneB`, then the path is set automatically to `/system/zones/zonename` based on the template.

Using the `zonecfg` Command

The `zonecfg info` command is used to display a zone's properties and resources. Use the following syntax:

```
global$ zonecfg -z zonename info [property] [-a]
```

Without the `-a` option, the command displays selected configured properties of the zone if you do not specify a specific property.

```
global$ zonecfg -z zoneA info
zonename: zoneA
brand: solaris
autoboot: true
net:
    physical=net1
anet:
    linkname: net0
    configure-allowed-address: true
    auto-mac-address: MACaddress
global$
```

If you add the `-a` option, then all the configurable properties of the zone are displayed, including those that have not been set.

```
global$ zonecfg -z zoneA info -a
zonename: zoneA
zonepath: /export/zones/zoneA
brand: solaris
autoboot: true
```

```

autosshutdow: shutdown
bootargs:
file-mac-profile:
pool:
limitpriv:
scheduling-class:
ip-type: exclusive
hostid:
hwprovider:
fs-allowed:
net:
    address:
    allowed-address:
    configure-allowed-address:
    physical=net1
    defrouter
anet:
    linkname: net0
    lower-link: auto
    allowed-address:
    configure-allowed-address: true
    defrouter:
    allowed-dhcp-cids:
    link-protection: mac-nospoof
    mac-address: auto
    auto-mac-address: MACaddress
    mac-prefix:
...
global$

```

You can display the same property information about a zone while in interactive configuration:

```

global$ zonecfg -z zoneA
zonecfg:zoneA> info [property]
                    selected configured properties

zonecfg:zoneA> info -a
                    all configurable properties

zonecfg:zoneA>

```

Using the `zlogin` Command

To display information about a zone's internal configuration, you would typically log in to the zone and then issue the appropriate Oracle Solaris commands. However, with `zlogin`, you can run these commands directly from the global zone. Use the following syntax:

```

global$ zlogin zonename
command

```

The following example shows the network IP configuration of `zoneA`.

```

global$ zlogin zoneA ipadm
NAME          CLASS/TYPE  UNDER  ADDR
lo0           loopback   ---    --
lo0/v4       static     ---    127.0.0.1/8
lo0/v6       static     ---    ::1/128
global$

```

The following example displays the computer or host name of `zoneA`.

```
global$ zlogin zoneA hostname
system1
```

The following example shows user home directories in `zoneA`.

```
global$ zlogin zoneA ls /export/home
jdoe
jsmith
```

For more information about the use of the `zlogin` command, see the [zlogin\(1\)](#) man page.

Installing Non-Global Zones Based on Existing Zones

You can take advantage of existing non-global zones on your system and use them as sources for installing additional zones. These sources can be used either through direct cloning or by using their Unified Archive files. This method shortens installation time considerably.

As shown in [How to Create and Deploy a Non-Global Zone](#), zone deployment involves configuration and installation processes. The following sections discuss how these processes are accomplished with source zones.

Using Clones

This method is appropriate for installing a small number of new zones and consists of two stages:

Exporting the source zone's configuration

All property settings of a zone are saved in storage specific to that zone. You extract the storage contents to a file to be applied to the new zones.

The source zone's configuration should ideally contain only minimal zone-specific settings so that adjustments to the exported file will also be minimal before you apply it to new zones. Application-specific configurations generally must be reconfigured in the clone. For example, if you have a database instance in a zone and you clone that zone, you might have to reconfigure the database instance in the clone so that it recognizes itself as a different instance.

Cloning the source zone

Cloning the source zone installs the new zone. At the end of this process, you can boot the zone and begin using it. Before you clone the source, ensure first that it is shut down.

Using Unified Archives

This method is useful for installing multiple zones for a large deployment.

A Unified Archive file contains images of a running Oracle Solaris system, including zones that are configured on it. Optionally, you can archive only the zones on the system for purposes of zone provisioning.

To create an archive, see [Creating a Unified Archive in Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4](#) and the [archiveadm\(8\)](#) man page.

To install zones from an archive, use the following command:


```
global$ zoneadm install -a full-path-to-archive [-z archived-zone] [-c full-path-to-profile.xml]
```

Specify the `-c` option if you are using a configuration profile. See [Using Manifests and Configuration Profiles](#) for information.

If the archive contains multiple zones, specify the `-z` option to select the zone to use.

 **Note:**

Support for `zoneadm` subcommands differ depending on the brand of zone being used. For details, refer to the [zoneadm\(8\)](#) man page.

See also [How to Deploy a Zone From a Unified Archive in Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4](#).

How to Install and Deploy Zones by Cloning

This procedure describes how to clone a source zone to install new zones.

1. **If necessary, shut down the source zone.**
2. **Export the zone's configuration to a file.**

```
global$ zonecfg -z source-zone export -f path-to-filename
```

3. **For every new zone you want to configure and install, do the following:**
 - a. **Edit the exported file so that any zone-specific parameter is set for the new zone.**

For example, the `zonepath`, if defined, would need to be changed.

- b. **Configure the new zone with the exported file.**

```
global$ zonecfg -z new-zone -f path-to-filename
```

- c. **Install the new zone by cloning the source zone.**

```
global$ zoneadm -z new-zone clone [-c full-path-to-profile.xml] source-zone
```

Specify the `-c` option if you are using a configuration profile. See [Using Manifests and Configuration Profiles](#) for information.

- d. **Boot the new zone.**

```
global$ zoneadm -z new-zone boot
```

- e. **If you did not use a configuration profile, manually configure the system information.**

Refer to [Step 5 of How to Create and Deploy a Non-Global Zone](#) for instructions.

Example 2-3 Creating Multiple Zones From a Source Zone

This example shows how to create three non-global zones from `zoneA`:

- Three zones to be created are `zone0`, `zone1`, and `zone2`.

- The `zonename` of `zoneA` is defined. Thus, this parameter needs to be adjusted for each new zone.
- Each zone has a corresponding configuration profile (`/var/tmp/zonename.xml`) to be used at installation time.

Shut down the zone if necessary.

```
global$ zoneadm -z zoneA shutdown
global$ zoneadm list -cv
ID NAME      STATUS      PATH                BRAND    IP
0  global     running    /                   solaris  shared
   zoneA     installed  /zones/zonea       solaris  exclusive
```

Export the configuration.

```
global$ zonecfg -z zoneA export -f /var/tmp/zoneA.txt
global$ cat /var/tmp/zoneA.txt
create -b
set brand=solaris
set zonename=/zones/zonea
set autoboot=true
```

Edit `/var/tmp/zoneA.txt` to set `zonename` to `/zones/zone0`. Then configure `zone0` with `/var/tmp/zoneA.txt`.

```
global$ zonecfg -z zone0 -f /var/tmp/zoneA.txt
global$ zonecfg -z zone0 info
zonename: zone0
brand: solaris
zonename: /zones/zone0specific to zone0
autoboot: true
```

For each of the remaining zone, repeat editing the `zonename` and configuring the zone. After all the new zones are configured, they will be listed as being in the `configured` state.

```
global$ zoneadm list -cv
ID NAME      STATUS      PATH                BRAND    IP
0  global     running    /                   solaris  shared
   zoneA     installed  /zones/zonea       solaris  exclusive
   zone0     configured /zones/zone0       solaris  exclusive
   zone1     configured /zones/zone1       solaris  exclusive
   zone2     configured /zones/zone2       solaris  exclusive
```

Clone the source zone to create the three zones.

```
global$ zoneadm -z zone0 clone -c /var/tmp/zone0.xml zoneA
global$ zoneadm -z zone1 clone -c /var/tmp/zone1.xml zoneA
global$ zoneadm -z zone2 clone -c /var/tmp/zone2.xml zoneA
```

```
global$ zoneadm list -cv
ID NAME      STATUS      PATH                BRAND    IP
0  global     running    /                   solaris  shared
   zoneA     installed  /zones/zonea       solaris  exclusive
   zone0     installed  /zones/zone0       solaris  exclusive
   zone1     installed  /zones/zone1       solaris  exclusive
   zone2     installed  /zones/zone2       solaris  exclusive
```

```
global$ zoneadm -z zone0 boot
global$ zoneadm -z zone1 boot
global$ zoneadm -z zone2 boot
```

```
global$ zoneadm list -cv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
	zoneA	installed	/zones/zonea	solaris	exclusive
1	zone0	running	/zones/zone0	solaris	exclusive
2	zone1	running	/zones/zone1	solaris	exclusive
3	zone2	running	/zones/zone2	solaris	exclusive

Creating a Source Zone

As an option, you can create a zone solely for use as a source to provision new zones. Configure the zone's global properties with settings that remain stable and universally applicable, such as `autoboot` and so on. You can also explicitly specify its brand so you know to use this source to install other zones of the same brand. Make sure to name the zone in a way to easily identify it as only a source zone. The zone *would not be booted* to prevent its being used.

Uninstalling and Deleting a Non-Global Zone

Both uninstalling a zone and deleting a non-global zone destroy the zone. If uninstalling is interrupted, you might be able to reconstruct the zone.

How to Uninstall and Remove a Zone

Ensure that all users are logged out of the zone.

This procedure combines three separate but dependent operations: shutting down a zone, uninstalling the zone, and deleting the zone.

Caution:

- Removing all files in the zone's root file system is irreversible.
- When you uninstall a zone that has its own ZFS file system for the `zonepath`, the ZFS file system is destroyed.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Verify that the zone is not in the running state.

```
global$ zoneadm list -cv
```

3. If the zone is running, shut down the zone.

- To shut down the zone immediately:

```
global$ zoneadm -z zonename shutdown
```

- To specify shutdown options:

```
global$ zlogin zonename shutdown [shutdown-options]
```

For example, to specify a 10 second wait period before shutting down:

```
global$ zlogin zonename shutdown -g 10
```

4. Verify that the zone is completely shut down.

```
global$ zoneadm list -cv
```

The zone to be deleted should have an `installed` status in the output.

5. Uninstall the zone.

```
global$ zoneadm -z zonename uninstall
```

6. Verify that the zone has been uninstalled.

```
global$ zoneadm list -cv
```

The zone to be deleted should have a `configured` status in the output.

7. To completely remove the zone from the host, delete the zone.

```
global$ zonecfg -z zonename delete [-F]
```

Use the `-F` option if you need to force the action. If you use this option, the command proceeds without issuing a confirmation request.

8. Verify that the deleted zone is no longer listed.

```
global$ zoneadm list -cv
```

Only the global zone should be listed in the output.

3

Managing Zone Resources and Properties

This chapter describes how to configure zone properties and resources. This chapter covers the following topics:

- [About Configuring Zone Properties](#)
- [Setting Values of Zone Properties and Resources](#)
- [Modifying Zone Property and Resource Settings](#)
- [Configuring the dataset Property on solaris Zones](#)
- [Performing Zone Configuration From the Global Zone](#)

For background information, review [Before You Begin Working With Oracle Solaris Zones](#).

About Configuring Zone Properties

Zone properties and resources together define a zone's characteristics as a virtual machine.

After a zone is created, you can begin to set its resources and properties, even before the zone is installed. However, the more typical approach is to complete property and resource configuration after installation. This approach reduces the risk of installation failures due to prior errors in configuration. See [How to Create and Deploy a Non-Global Zone](#).

When you are configuring a resource, the zone prompt identifies that resource scope, such as `fs` for file system properties, `anet` for automatic network interface properties, and so on.

Zone configuration has two modes:

Default

Use this mode for most of your configuration tasks. It enables you to configure persistent settings for any zone property and resource. The settings take effect when you boot or reboot the zone. However, for a zone that is already running, you can apply the settings without rebooting by issuing the `zoneadm apply` command.

Live

This mode can be used only on a running zone, and only to a subset of zone properties and resources, depending on the zone's brand. To know which resources and properties can be configured in this mode, refer to the zone brand's corresponding man page.

The changes take effect immediately after you commit them, and no zone reboot is necessary. However, the changes remain active only until the next zone reboot. Essentially, you use the `live` mode to make temporary changes.

▲ Caution:

The `zoneadm apply` command behaves differently in this mode than what you might expect. This command loads the configuration that is in storage and cancels the active settings that you just committed. Thus, in `live` mode, you use `zoneadm apply` to *nullify your temporary changes*, if you do not want to reboot the zone.

As you set properties, the values are stored in memory. When you exit a zone configuration session, the settings are saved to stable storage and take effect according to the mode of configuration you use.

For more information, see the `zonecfg(8)` man page.

How to Configure Zone Properties and Resources

To configure and manage a non-global zone's resources and properties, you use `zonecfg` subcommands. Whether you do the configuration before or after the zone is installed, the syntax and sequence of steps remain the same.

✎ Note:

For clarity, this procedure uses the `default` mode.

- 1. Become a zone administrator.**

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

- 2. Switch to interactive configuration.**

```
global$ zonecfg -z zonename
zonecfg:zonename>
```

- 3. For every resource, use the add, set, and end subcommands in sequence to configure the resource's properties.**

```
zonecfg:zonename>
add resource
zonecfg:zonename:resource> set property=value
zonecfg:zonename:resource> end
```

Use the `set` subcommand repeatedly as needed to configure different properties of the resource. Use the `end` subcommand to end the resource configuration and return to the zone's main prompt.

- 4. For properties that do not belong to any resource, set them from the zone's main prompt.**

```
zonecfg:zonename> set property=value
zonecfg:zonename>
```

You can set multiple properties and resources in a single configuration session. Proceed to the next step if you want to terminate the session.

- 5. Return to the global zone.**

```
zonecfg:zonename> exit
global$
```

Before the `exit` subcommand returns you to the global zone, it first performs the actions of two separate subcommands which you can also independently run: `verify` which checks that the settings are valid and `commit` which saves the settings to stable storage.

6. Apply the settings to the zone.

Choose one of the following methods:

- Issue the `zoneadm apply` command.

You can use this command only if the zone is running. With this command, no zone reboot is required.

```
global$ zoneadm -z zonename apply
Checking: Adding resource
property=setting
Applying the changes.
```

- Reboot the zone.

```
global$ zoneadm -z zonename reboot
```

Setting Values of Zone Properties and Resources

This section contains examples of how to set selected zone properties and resources. All the examples assume the following:

- You have been granted the appropriate rights or role.
- You are in a profile shell.
- You apply the settings either through the `zoneadm apply` command or by rebooting the zone.

Note:

The examples here are not exhaustive. For a complete list of zone resources and properties, see the [zonecfg\(8\)](#) man page or [Oracle Solaris Zones Configuration Resources](#).

The examples also show how to display the configurations that you set by using the `zonecfg info` syntax as described in [Using the zonecfg Command](#).

Limiting the Zone's Resources

Limiting the resources that a zone can use helps you to manage resource usage between the non-global zone and the global zone, or among multiple zones on the host. You can better allocate resources among the different zones so that each zone has only what it needs to fulfill its purpose.

Assigning CPUs to the Zone

This example shows how to dedicate specific CPUs or assign a range of CPUs for the zone to use.

```

global$ zonecfg -z zoneA
zonecfg:zoneA> add dedicated-cpu
zonecfg:zoneA:dedicated-cpu> set ncpus=1-2
zonecfg:zoneA:dedicated-cpu> set importance=10
zonecfg:zoneA:dedicated-cpu> end
zonecfg:zoneA> info
...
dedicated-cpu:
    ncpus: 1-2
    importance: 10
...
zonecfg:zoneA> exit
global$ Reboot or run zoneadm apply command.

```

Example 3-1 Making Persistent Changes to a Running Zone

This example specifies `zonecfg` commands to set the `cpu-shares` resource, clear the `pool` resource settings, and add an `anet` resource for the running zone `zone1`, then applies the changes to the persistent zone configuration.

```

global$ pfbash zonecfg -z zone1 "set cpu-shares=4;clear pool;add anet;"
global$ zoneadm -z zone1 apply
zone1: Checking: set property cpu-shares=4
zone1: Checking: clear property pool
zone1: Checking: add anet linkname=myanet0
zone1: Applying changes

```

Setting Cap Limits for Memory

This example shows how to set limits for memory, swap memory, and locked memory.

```

global$ zonecfg -z zoneA
zonecfg:zoneA> add capped-memory
zonecfg:zoneA:capped-memory> set physical=1g
zonecfg:zoneA:capped-memory> set swap=2g
zonecfg:zoneA:capped-memory> set locked=500m
zonecfg:zoneA:capped-memory> end
zonecfg:zoneA> info
...
capped-memory:
    physical: 1G
    [swap: 2G]
    [locked: 500M]
rctl 0:
    name: zone.max-swap
    value: (priv=privileged,limit=2147483648,action=deny)
rctl 1:
    name: zone.max-locked-memory
    value: (priv=privileged,limit=524288000,action=deny)
...
zonecfg:zoneA> exit
global$ Reboot or run zoneadm apply command.

```

Adding File Systems for Zone Use

You can add multiple file systems for zone use through the `fs` resource. This resource has multiple file system types. The following examples show how to use the LOFS and UFS file types. For examples of using ZFS, the default file system type, see [Configuring the dataset Property on solaris Zones](#).

Example 3-2 Sharing a Directory With the Global Zone

This example specifically involves the type called loopback file system (`lofs`) which enables a zone to share a directory with the global zone. The `lofs` file system type is also used to grant zone access to media such as a DVD drive.

Suppose that you want to grant zone access to the `/opt/mysoftware` directory in the global zone. In the example's different commands, `special` identifies the global directory to be shared, `dir` specifies the mount point in the zone through which the global directory is accessed, and `options` specifies the zone permissions (read-only) in the shared directory. The read-only permission prevents users in the non-global zone from making changes to the global zone directory.

```
global$ zonecfg -z zoneA
zonecfg:zoneA> add fs
zonecfg:zoneA:fs> set type=lofs
zonecfg:zoneA:fs> set special=/opt/mysoftware
zonecfg:zoneA:fs> set dir=/usr/software
zonecfg:zoneA:fs> set options=ro
zonecfg:zoneA:fs> end
zonecfg:zoneA> info
...
fs 0:
    dir: /usr/software
    special: /opt/mysoftware
    raw not specified
    type: lofs
    options: [ro]
...
zonecfg:zoneA> exit
global$ Reboot or run zoneadm apply command.
```

Example 3-3 Assigning a UNIX File System to a Zone

The `fs` resource also supports UNIX file systems (UFS). Suppose that you have a UFS file system on the slice `/dev/rdisk/c1t1d0s0`. This example shows how to make that file system available to the zone.

In the example, `special` identifies the block device path of the UFS file system, `raw` is the raw device path where the `fsck` process is run, and `dir` is the mount point of the file system for the zone.

```
global$ zonecfg -z zoneA
zonecfg:zoneA> add fs
zonecfg:zoneA:fs> set type=ufs
zonecfg:zoneA:fs> set special=/dev/dsk/c1t1d0s0
zonecfg:zoneA:fs> set raw=/dev/rdisk/c1t1d0s0
zonecfg:zoneA:fs> set dir=/data
zonecfg:zoneA:fs> end
zonecfg:zoneA> exit
global$ zonecfg -z zoneA info
...
fs 0
    dir: /usr/software
    special: /opt/mysoftware
    raw not specified
    type: lofs
    options: [ro]

fs 1
```

```

dir: /data
special: /dev/dsk/ctl1d0s0
raw: /dev/rdisk/ctl1d0s0
type: ufs
options: [ ]
...
global$Reboot or run zoneadm apply command.

```

Setting the `hwprovider` Property to Override the Global Zone Value (`solaris10` Only)

The `hwprovider` property indicates the current hardware manufacturer string. To determine the value in a `solaris10` branded zone, review the output of the `prtconf` command or the `sysinfo(SI_HW_PROVIDER, ..)` system call. You can also use the `zonecfg -z solaris10brandedzonename info -a` command.

Use the `zonecfg` command to set or clear the string in the zone.

- To set the string in the zone, use the following command:

```
zonecfg> set hwprovider = "Sun_Microsystems"
```

- To clear the string and use the global zone string:

```
zonecfg> clear hwprovider
```

For more information, see [hwprovider Global Property \(solaris10 Only\) in Oracle Solaris Zones Configuration Resources](#).

Setting Time Values in Non-Global Zones

You can set time values in non-global zones that are different from the value in the global zone by using the Network Time Protocol (NTP). Two features interact, the `global-time` property for the zone and the `sys_time` privilege on the time-setting process. While you can set different times in non-global zones, if the time is changed in the global zone, the non-global zones time are offset by the same amount.

NTP can be run from any zone, affecting only the zone in which the command is run. When running NTP across a system with non-global zones that have different times, run NTP in the global zone to synchronize all the non-global zone clocks that just run at an offset. The effect of NTP changing the time through clock modulation in the global zone will transfer to a non-global-zone as well.

- When NTP is run inside the zone with `global-time` set to `false`, the `ntp_adjtime` and `adjtime` system calls cannot be used to make corrections to the zone time. When `global-time` is set to `false`, NTP synchronizes the time by adjusting the clock to a given value.
- When NTP is run inside the zone with `global-time` set to `true`, NTP can modulate the system clock through the `ntp_adjtime` and `adjtime` system calls. The `true` setting preserves keeping system time in tight synchronization with the zone.

For more information about the `global-time` property, see [global-time Global Property \(solaris and solaris10 Only\) in Oracle Solaris Zones Configuration Resources](#). For information about setting the time, see the `date(1)` man page. For more information about privileges, see [Privileges in a Non-Global Zone](#). For more information about `adjtime` and `ntp_adjtime`, see the `adjtime(2)` and `ntp_adjtime(2)` man pages.

Modifying Zone Property and Resource Settings

Modifying zone property and resource settings is generally similar to adding resources. The following section first discusses how, if the situation requires, you can abort a configuration session and safely exit to keep your zone's current settings intact.

Canceling a Configuration Session

Two `zonecfg` subcommands enable you to halt the process any time while you are modifying property and resource settings. These subcommands can be used in both `default` and `live` modes of property configuration:

cancel Subcommand

This command discards the changes you just made to properties of a resource. The command applies only to resource configuration.

```
zonecfg:zonename:  
resource> resource configuration commands  
zonecfg:zonename:  
resource> cancel  
zonecfg:zonename>
```

Canceling a resource reconfiguration nullifies the changes you set only for that resource. If you exit the `zonecfg` session, other changes you made to other properties can still apply.

reload Subcommand

The command discards any uncommitted changes and retrieves the configuration from a stable storage. You can issue this command only from the main zone prompt.

```
zonecfg:zonename:> configuration commands  
zonecfg:zonename:  
resource> resource configuration commands  
zonecfg:zonename:  
resource> end  
...  
zonecfg:zonename> reload  
Are you sure you want to reload (y/[n])?
```

This command returns the zone to its original status at the beginning of the `zonecfg` session before you began configuring properties. See [Canceling a Modification and Restoring Saved Settings](#).

How to Modify Zone Resources and Properties

If you plan to modify zone properties in `live` mode, ensure that the properties are supported for configuration in this mode. Otherwise, unexpected results might occur.

- 1. Become a zone administrator.**

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

- 2. (Optional) If you plan to configure in live mode, check that the zone is running.**

```
global$ zoneadm list -cv
```

- 3. (Optional) Display the current settings of the zone.**

```
global$ zonecfg -z zonename info
```

Through this step, you can identify the property or resource you want to change.

4. Switch to interactive configuration.

```
global$ zonecfg -z zonename [-r]
```

Use the `-r` option to reconfigure in `live` mode.

5. Select the resource type to be changed.

```
zonecfg:zonename> select resource [identifier]
```

A zone might have multiple settings for the same resource type, as illustrated in [Adding File Systems for Zone Use](#). Make sure you are modifying the correct resource type.

6. Modify the setting.

The subcommands you use depend on what type of modification you want to make.

- To replace a current setting with a new setting:

```
zonecfg:zonename:resource> remove (value-pair1[,value-pair2,...])  
zonecfg:zonename:resource> add (value-pair1[,value-pair2,...])
```

- To reset a property to the default value:

```
zonecfg:zonename:resource> clear property
```

- To remove the resource:

```
zonecfg:zonename:resource> delete
```

Caution:

The deletion occurs instantaneously on both in-memory and stable storage and is irreversible.

7. If you are within a resource scope, end the modification of the resource.

```
zonecfg:zonename:resource> end
```

8. Return to the global zone.

```
zonecfg:zonename> exit
```

If you are modifying properties in `live` mode, the changes take effect immediately when you exit. However, these changes are active only up to the next zone reboot. If you reboot the zone, the previous settings in storage are restored.

9. If you are modifying properties in default mode, apply the changes.

Choose one of the following methods:

- Issue the `zoneadm apply` command.

You can use this command only if the zone is in a running state. With this command, no zone reboot is required.

▲ Caution:

Running this command after configuring in `live` mode will cancel the changes you just made. See caution in [About Configuring Zone Properties](#).

```
global$ zoneadm -z zonename apply
Checking: Adding resource
property=setting
Applying the changes.
```

- Reboot the zone.

```
global$ zoneadm -z zonename reboot
```

Example 3-4 Changing the Settings of a Resource Control

A resource control is represented in zone configuration as `rctl`. In this example, multiple resource controls are configured in a zone and you are modifying `zone.cpu-shares`. The change is applied to the zone without rebooting through the `zoneadm apply` command.

```
global$ zonecfg -zoneA
zonecfg:zoneA> info
...
rctl 1
    name: zone.max-locked-memory
    value: (priv=privileged,limit=524288000,action=deny)
rctl 2
    name: zone.cpu-shares
    value: (priv=privileged,limit=20,action=none)
...
zonecfg:zoneA> select rctl 2
zonecfg:zoneA:rctl> info
rctl 3
    name: zone.cpu-shares
    value: (priv=privileged,limit=20,action=none)
zonecfg:zoneA:rctl> remove value (priv=privileged,limit=20,action=none)
zonecfg:zoneA:rctl> add value (priv=privileged,limit=30,action=none)
zonecfg:zoneA:rctl> info
rctl 2
    name: zone.cpu-shares
    value: (priv=privileged,limit=30,action=none)Value of limit is changed.
zonecfg:zoneA:rctl> end
zonecfg:zoneA> exit
global$ zoneadm -z zoneA apply
Checking: Adding rctl value (priv=privileged,limit=30,action=none)
Applying the changes.
```

Example 3-5 Reverting a Zone's autoboot Property to the Default Setting

By default, a zone does not boot whenever the host system is rebooted. In this example, you are resetting `zoneA`'s `autoboot` property to the default setting. The change takes effect after the zone is rebooted.

```
global$ zonecfg -z zoneA
zonecfg:zoneA> info autoboot
autoboot: true
zonecfg:zoneA> clear autoboot
zonecfg:zoneA> info autoboot
autoboot: false
```

```
zonecfg:zoneA> exit
global$ zoneadm -z zoneA reboot
```

Example 3-6 Canceling a Modification and Restoring Saved Settings

This example shows how to discard changes you just made and revert to the settings currently in storage.

```
global$ zonecfg -z zoneA
zonecfg:zoneA> info
zonename: zoneA
brand: solaris
autoboot: true
...
dedicated-cpu:
    ncpus: 1-2
    importance: 10
...
zonecfg:zoneA> remove dedicated-cpu
zonecfg:zoneA> info
zonename: zoneA
brand: solaris
autoboot: true
...
Dedicated-cpu no longer listed.
...
zonecfg:zoneA> reload
Are you sure you want to reload (y[n])? y
zonecfg:zoneA> info
zonename: zoneA
brand: solaris
autoboot: true
...
dedicated-cpu:Dedicated-cpu is restored.
    ncpus: 1-2
    importance: 10
...
zonecfg:zoneA> exit
global$ zoneadm -z zoneA reboot
```

Example 3-7 Temporarily Modifying a Configured Property by Using live Mode

This example shows the processes in live mode configuration. In this scenario, zoneA has a second anet resource configuration that you want to remove. In the solaris brand, anet is a resource that can be reconfigured in live mode.

The example begins by verifying that zoneA is running so that live mode can be used.

```
global$ zoneadm list -cv
ID NAME      STATUS      PATH                                BRAND  IP
0  global     running     /                                    solaris shared
1  zoneA      running     /export/zones/zoneA               solaris exclusive
-  zoneB      installed   /system/zones/zoneB               solaris exclusive

global$ zonecfg -z zoneA info
...
anet 0
    name: net0
    configure-allowed-address: true
    auto-mac-address: 2:8:20:af:4c:0
anet 1
```

```
name: net1
configure-allowed-address: true
auto-mac-address: 2:8:20:1:9b:31
...
global$ zonecfg -z zoneA -r
zonecfg:zoneA> remove anet linkname=net1
zonecfg:zoneA> info
...
anet 0
name: net0
configure-allowed-address: true
auto-mac-address: 2:8:20:af:4c:0
...anet 1 no longer listed.

zonecfg:zoneA> exit
Checking: Removing anet linkname=net1
Applying the changes
```

From the global zone, you can display the *temporary* changes as follows:

```
global$ zonecfg -z zoneA -r info
```

Rebooting the zone cancels the temporary changes. To cancel without rebooting, use the `zoneadm apply` command:

```
global$ zoneadm -z zoneA apply
Checking: Adding anet linkname=net1
Applying the changes

global$ zonecfg -z zoneA info
...
anet 0
name: net0
configure-allowed-address: true
auto-mac-address: 2:8:20:af:4c:0
anet lanet 1 is restored.
name: net1
configure-allowed-address: true
auto-mac-address: 2:8:20:1:9b:31
...
```

Configuring the `dataset` Property on `solaris` Zones

Non-global zones use the ZFS file system as their Oracle Solaris hosts do. Although ZFS is supported as an `fs` type, you typically use the `dataset` property instead of the `fs` property when configuring non-global zones' datasets. The `dataset` property enables you to have control over the dataset from within the zone, where you can access the dataset and configure ZFS properties as needed.

You can add, export, and remove datasets from non-global zones while the zones are running. You can change a snapshot of the zone by using *live* mode, or modify the zone in *default* mode, which affects the zone that is running on the system. For a fuller explanation, see [About Configuring Zone Properties](#). The `zonecfg(8)` man page describes the `-r` option for configuring a zone in live mode.

The following examples show how to add a dataset, remove a dataset, and remove a dataset by exporting it. Export operations precede migrating a zone.

Example 3-8 Adding a Dataset to a Zone

Suppose that you created a ZFS pool called `rpool/zonefs`. To add this pool to the non-global zone using the default mode, you would do the following steps:

```
global$ zonecfg -z zoneA
zonecfg:zoneA> add dataset
zonecfg:zoneA:fs> set name=rpool/zonefs
zonecfg:zoneA:fs> end
zonecfg:zoneA> exit
global$ zonecfg -z zoneA info
...
dataset
  name: rpool/zonefs
  alias: zonefs
...
```

Apply the changes.

```
global$ zoneadm -z zoneA apply
```

You can now run `zfs` subcommands on the dataset.

Example 3-9 Removing a Dataset From a Running Zone

1. The administrator starts by viewing the dataset information of a running zone from the global zone.

```
global$ pfbash zonecfg -z ZoneA -r info dataset
...
dataset:
  name: rpool/export/zonefs1
  alias: zonefs1
dataset:
  name: rpool/export/zonefs2
  alias: zonefs2
...
```

2. The administrator then removes the `zonefs1` dataset from the zone snapshot.

```
global$ zonecfg -z ZoneA -r
zonecfg:ZoneA> remove dataset name=rpool/export/zonefs1
zonecfg:ZoneA> info dataset
dataset 0:
  name: rpool/export/zonefs2
  alias: zonefs2
...
zonecfg:ZoneA> commit
zone 'ZoneA': Checking: Removing dataset name=rpool/export/zonefs1
zone 'ZoneA': Applying the changes
zonecfg:ZoneA> exit
```

3. The administrator verifies that the dataset is removed.

```
global$ zonecfg -z ZoneA -r info dataset
dataset:
  name: rpool/export/zonefs2
  alias: zonefs2
```

When ZoneA is rebooted, or when you use the `zoneadm apply` command, the dataset is restored to the zone. The `-r` option is for temporary changes only.

Example 3-10 Exporting a Dataset From a solaris Zone

In this example, the administrator exports a dataset for migration or move. If the dataset is in use, the export might not succeed. For more information, see [Exporting a ZFS Storage Pool in Managing ZFS File Systems in Oracle Solaris 11.4](#).

The administrator exports the dataset from the snapshot of the zone. At reboot, the dataset is still in the zone.

1. From the global zone, the administrator displays the zone datasets.

```
global$ pfbash zonecfg -z ZoneA info dataset
...
dataset:
    name: rpool/export/zonefs1
    alias: zonefs1
dataset:
    name: rpool/export/zonefs2
    alias: zonefs2
...
```

2. From the global zone, the administrator exports the zonefs2 dataset.

```
global$ zpool export zonefs2
```

3. The list of running zpools does not include the exported dataset.

```
global$ zpool list
NAME      SIZE  ALLOC   FREE   CAP  DEDUP  HEALTH  ALTROOT
zonefs1   68G   21.5G  46.5G  31%  1.00x  ONLINE  -
rpool     68G   21.5G  46.5G  31%  1.00x  ONLINE  -
```

Performing Zone Configuration From the Global Zone

Using the `zonecfg` tool interactively is the safest way to configure zone properties and resources. As an alternative, you can configure zones directly from the global zone with the following syntax:

```
global$ zonecfg -z zonename "zonecfg-subcommands"
```

Separate zone subcommands with semi-colons.

In the following example, `zoneA` is being configured with 4 shared CPUs and a new `anet` resource. At the same time, the zone's ZFS pool is being reset to its default value. The configuration is being performed in `default` mode and the changes are persistent.

```
global$ zonecfg -z zoneA "set cpu-shares=4;clear pool;add anet;set linkname=net2;end"
global$ zoneadm -z zoneA apply
Checking: Setting property cpu-shares=4
Checking: Clearing property pool
Checking: Adding anet linkname=net2
Applying the changes
global$
```

This example shows how to make temporary changes to a zone's setting by using the `live` mode.

```
global$ zonecfg -z zoneA -r \
"remove anet linkname=net2;add dataset;set name=rpool/zonefs;end"
Checking: Removing anet linkname=net2
Checking: Adding dataset name=rpool/zonefs
```

Applying the changes
global\$

Rebooting the zone restores the zone to its original settings.

 **Note:**

Direct configuration from the global zone is risky because errors can easily occur when you type lengthy commands. Use this method with caution and only if you thoroughly understand the effects of `zonecfg` subcommands.

4

Administering Networking in Oracle Solaris Non-Global Zones

This chapter covers general administration tasks and provides usage examples.

- [About Networking In Zones](#)
- [Using IP Network Multipathing With Non-Global Zones](#)
- [Managing Datalinks Related to Zones](#)
- [Changing Existing Shared-IP Zones to Exclusive-IP Zones](#)

See [About Non-Global Zone Administration](#) for general zone administration topics.

About Networking In Zones

By combining Oracle Solaris zones together with Oracle Solaris network virtualization, you can create networked zones that reduce the system footprint at your site that would otherwise be large if you use physical systems. For an illustration, see https://download.oracle.com/otndocs/tech/OTN_Demos/data-center-consolidation.html.

For more information about Oracle Solaris networking, refer to the following resources:

- [Managing Network Virtualization and Network Resources in Oracle Solaris 11.4](#)
- [Configuring and Managing Network Components in Oracle Solaris 11.4](#)

How to Create a Private Network of Multiple Zones

This procedure summarizes the steps to set up a private network of zones.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Create the etherstub and VNICs on the global zone.

Use the appropriate `dladm` subcommands. See the `dladm(8)` man page or [How to Configure VNICs and Etherstubs in Managing Network Virtualization and Network Resources in Oracle Solaris 11.4](#).

3. Configure and install the zones of the private network.

To complete this step quickly by using a source zone, see [How to Install and Deploy Zones by Cloning](#).

4. Configure the zones' physical property.

To configure interactively, see [How to Configure Zone Properties and Resources](#).

5. Boot the zones.

```
global$ zoneadm -z zonename boot
```

6. Configure the zones' VNICs with IP addresses.

For this step, you would typically log in to each zone.

```
global$ zlogin -C zonename
```

Then you use appropriate `ipadm` subcommands to configure the VNICs. For reference, see the `ipadm(8)` man page or [How to Configure an IPv4 Interface in Configuring and Managing Network Components in Oracle Solaris 11.4](#).

Example 4-1 Setting Up a Private Network

This example goes through the process of setting `zone0`, `zone1`, and `zone2` as a private network. The scenario consists of the following:

- The global zone's IP address is 192.0.2.100
- The private network of the zones is 192.0.1.0.
- The zones (`zone0`, `zone1`, and `zone2`) will be configured with VNICs (`vnic0`, `vnic1`, and `vnic2`, respectively).

This example assumes that the zones have already been created and installed.

For demonstration purposes, instead of using `zonecfg` interactively, zone configuration will be performed on the global zone as described in [Performing Zone Configuration From the Global Zone](#). Likewise, IP configuration will be performed from the global zone instead of from within each zone, as described in [Using the `zlogin` Command](#).

```
Prepare the etherstub and VNICs on the host.
global$ dladm create-etherstub stub0
global$ dladm create-vnic -l stub0 vnic0
global$ dladm create-vnic -l stub0 vnic1
global$ dladm create-vnic -l stub0 vnic2
                                Set the zones' physical property.
global$ zonecfg -z zone0 "add net;set physical=vnic0; end"
global$ zonecfg -z zone1 "add net;set physical=vnic1; end"
global$ zonecfg -z zone2 "add net;set physical=vnic2; end"
                                Boot the zones.
global$ zoneadm -z zone0 boot
global$ zoneadm -z zone1 boot
global$ zoneadm -z zone2 boot
                                Configure the VNICs with IP addresses.
global$ zlogin zone0 ipadm create-ip vnic0
global$ zlogin zone0 ipadm create-addr -a 192.0.1.100/24 vnic0

global$ zlogin zone1 ipadm create-ip vnic1
global$ zlogin zone1 ipadm create-addr -a 192.0.1.101/24 vnic0

global$ zlogin zone2 ipadm create-ip vnic2
global$ zlogin zone2 ipadm create-addr -a 192.0.1.102/24 vnic0
                                Display all the information.
global$ dladm
LINK          CLASS      MTU    STATE    OVER
...
stub0         etherstub  9000   unknown  --
vnic0         vnic       9000   up        stub0
vnic1         vnic       9000   up        stub0
vnic2         vnic       9000   up        stub0
zone0/vnic0   vnic       9000   up        stub0
zone1/vnic1   vnic       9000   up        stub0
zone2/vnic2   vnic       9000   up        stub0
```

```

global$ zlogin zone0 ipadm
NAME          CLASS    TYPE/STATE  UNDER  ADDR
...
vnic0         ip       ok          --     --
  vnic0/v4    static  ok          --     192.0.1.100/24

global$ zlogin zone1 ipadm
NAME          CLASS    TYPE/STATE  UNDER  ADDR
...
vnic1         ip       ok          --     --
  vnic1/v4    static  ok          --     192.0.1.101/24

global$ zlogin zone2 ipadm
NAME          CLASS    TYPE/STATE  UNDER  ADDR
...
vnic2         ip       ok          --     --
  vnic2/v4    static  ok          --     192.0.1.102/24

```

To enable communications between the private network (192.0.1.0) and the external network (192.0.2.0), configure one of the zones with another port that will provide the connectivity. Enable IP forwarding on that port. For this part of the example, interactive configuration is used.

```

global$ zonecfg -z zone2
zonecfg:zone2> add net
zonecfg:zone2:net> set physical=net1
zonecfg:zone2:net> end
zonecfg:zone2> exit
global$ zoneadm -z zone2 apply
Checking: Adding net physical=net1
Applying the changes

global$ zlogin zone2
[Connected to zone 'zone2' pts/3]
username@zone2:~$ ipadm create-ip net1
username@zone2:~$ ipadm create-addr -a 192.0.2.110 net1
username@zone2:~$ ipadm
NAME          CLASS    TYPE/STATE  UNDER  ADDR
...
net1          ip       ok          --     --
  net1/v4     static  ok          --     192.0.2.110/24

vnic2         ip       ok          --     --
  vnic2/v4    static  ok          --     192.0.1.102/24
username@zone2:~$ ipadm set-prop forwarding=on ipv4
username@zone2:~$ exit
logout

[Connection to zone 'zone2' pts/3 closed]
global$ zoneadm -z zone2 reboot

```

Test the connections by pinging a zone from a system in the 192.0.2.0 network, or conversely, pinging the 192.0.2.0 network from inside a zone.

Using IP Network Multipathing With Non-Global Zones

IP Network Multipathing (IPMP) can be configured on an exclusive-IP zone just as it is configured on the global zone.

For background information about IPMP, see [Configuring IPMP Groups in Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4](#). On an exclusive-IP type zone, you configure IPMP within the zone. The steps are similar to configuring IPMP on a global zone.

For shared-IP type zones, you can extend IPMP support from the global zone to the non-global zone by performing the following procedure.

How to Extend IP Network Multipathing Functionality to Shared-IP Non-Global Zones

Use this procedure to configure IPMP in the global zone and extend the IPMP functionality to non-global zones.

Each address, or logical interface, should be associated with a non-global zone when you configure the zone. See [zonecfg and the Global Zone in Oracle Solaris Zones Configuration Resources](#) and [How to Create and Deploy a Non-Global Zone](#) for instructions.

This procedure accomplishes the following:

- The cards `net0` and `net1` are configured together in an IPMP group whose interface is `ipmp0`.
- A data address of `ipmp0`, `192.0.2.0`, is associated with the non-global zone `my-zone`.
- The `net0` card is set as the zone's physical interface.

In a running zone, you can use the `ipadm` command to make the association. See [Shared-IP Network Interfaces](#) and the `ipadm(8)` man page for more information.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. In the global zone, configure IPMP groups.

Follow procedures in [Configuring IPMP Groups in Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4](#).

3. Use the `zonecfg` command to configure the zone.

When you configure the `net` resource, add address `192.0.2.0` and physical interface `net0` to the zone `zonename`:

```
zonecfg:my-zone> add net
zonecfg:my-zone:net> set address=192.0.2.0
zonecfg:my-zone:net> set physical=net0
zonecfg:my-zone:net> end
```

Only `net0` would be visible in the non-global zone `my-zone`.

If `net0` Subsequently Fails

If `net0` subsequently fails, because `192.0.2.0` is assigned to `ipmp0`, then that address continues to be available through `net1`. The availability of the address also applies to `my-zone`. However, after `net0` has failed, then `net1` becomes the visible interface of `my-zone`.

Managing Datalinks Related to Zones

The `dladm` command supports a property called `zone` that enables you to display information about datalinks that are zone-related:

```
global$ dladm show-linkprop -p zone
LINK          PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
...
vnic0         zone      rw    zone0  --         --       --
zone0/vnic0   zone      rw    zone0  --         --       --
...
```

You can temporarily assign a specific zone to a link with the following command:

```
global$ dladm set-linkprop -t -p zone=zone0 vnic1
global$ dladm show-link -z zone0
LINK          CLASS  MTU  STATE  OVER
zone0/vnic0   vnic   9600  up     stub0
zone0/vnic1   vnic   9600  up     stub0
...
```

To remove the temporary setting, reboot the zone.

Tip:

If the running zone is using the device, the reassignment fails and an error message is displayed. See [Exclusive-IP Zone Is Using Device, so dladm reset-linkprop Fails](#).

Tip:

If the running zone is using the device, the reassignment fails and an error message is displayed. See [Exclusive-IP Zone Is Using Device, so dladm reset-linkprop Fails](#).

Changing Existing Shared-IP Zones to Exclusive-IP Zones

Shared-IP type zones can be updated to exclusive-IP zones. In most cases, you replace the `zonecfg net` resources with `anet` resources and change the `ip-type` to `exclusive`.

If you have existing shared-IP zones using interfaces that are part of a global zone IPMP group, switch to using DLMP aggregations. Use the `dladm create-aggr` subcommand to create the aggregations.

1. In the global zone, create a DLMP aggregation on the old IPMP interfaces.
2. Create a `zonecfg anet` resource, where the `lower-link` property points to the DLMP aggregation.

For information about `dladm`, see the `dladm(8)` man page. Also see [Strategies for Network Administration in Oracle Solaris 11.4](#).

5

Administering SMF and Reporting in Oracle Solaris Non-Global Zones

This chapter covers general administration tasks and provides usage examples.

- [Zones and the Service Management Facility](#)
- [Administering the Zones Delegated Restarter](#)
- [Reporting the Status of SMF Services in a Non-Global Zone](#)
- [Reporting Resource Usage in a Non-Global Zone](#)
- [Reporting Per-Zone File System Statistics](#)
- [Using the Fair Share Scheduler on a System With Zones Installed](#)

See [About Non-Global Zone Administration](#) for general zone administration topics.

Zones and the Service Management Facility

The Oracle Solaris Service Management Facility (SMF) framework manages system and application services. SMF manages critical system services essential to the working operation of the system and also manages application services such as a database or web server. SMF improves the availability of a system by ensuring that essential system and application services run continuously, even in the event of a hardware or software failure.

The zones SMF service has mechanisms for prioritizing zones booting order, setting a threshold for the number of zones booting in parallel, and providing integration with Fault Management Architecture (FMA).

Zone SMF Dependencies

The set of Oracle Solaris Service Management Facility (SMF) Fault Management Resource Identifiers (FMRIs) that establish the dependencies for the zone are defined in the `smf-dependency` resource. The dependencies are comprised of `name`, `fmri`, and `grouping` properties. The dependencies of the `svc:/system/zones:default` service are as follows:

require_all Dependencies

```
svc:/system/filesystem/local:default
svc:/milestone/name-services:default
svc:/milestone/devices:default
svc:/milestone/network:default
```

optional_all Dependencies

```
svc:/system/pools:default
svc:/system/zones-monitoring:default
```

The set of SMF FMRIs that make up the zone dependencies is defined in the `smf-dependency` resource in the `zonecfg` command.

For more information about setting dependencies, see [Chapter 1, Non-Global Zone Configuration Command and Resources in Oracle Solaris Zones Configuration Resources](#).

Example 5-1 Setting SMF Dependencies on a Zone

Dependency descriptions:

- `require_all: svc:/application/list:default`
- `require_any: svc:/system/zones/zone:appfirewall`
- `require_any: svc:/3rdparty/my-firewall:default`
- `exclude_all: svc:/system/zones/zone:dataload`

```
global$ pfbash zonecfg -z my-zone
zonecfg:my-zone> add smf-dependency
zonecfg:my-zone:smf-dependency> set fmri=svc:/application/list:default
zonecfg:my-zone:smf-dependency> end
zonecfg:my-zone> add smf-dependency
zonecfg:my-zone:smf-dependency> set name=firewall
zonecfg:my-zone:smf-dependency> set fmri=svc:/system/zones/zone:appfirewall
zonecfg:my-zone:smf-dependency> set grouping=require_any
zonecfg:my-zone:smf-dependency> end
zonecfg:my-zone> add smf-dependency
zonecfg:my-zone:smf-dependency> set name=firewall
zonecfg:my-zone:smf-dependency> set fmri=svc:/3rdparty/my-firewall:default
zonecfg:my-zone:smf-dependency> end
zonecfg:my-zone> add smf-dependency
zonecfg:my-zone:smf-dependency> set fmri=svc:/system/zones/zone:dataload
zonecfg:my-zone:smf-dependency> set grouping=exclude_all
zonecfg:my-zone:smf-dependency> end
zonecfg:my-zone> exit
```

For more information about the `smf-dependency` resource, see [Oracle Solaris Zones Configuration Resources](#) and the `zonecfg(8)` man page.



Note:

Failures during the infrastructure setup of a particular zone will place that zone in the maintenance state. Use the `svcadm clear` command on a zone instance to cause the zones delegated restarter to retry failed `zoneadm boot` or `zoneadm attach` operations for that zone.

About the Zone SMF Instance Service State

The zone SMF instance service state is determined by the zone `milestone/goals` state. The `milestone/goals` goal service provides an unambiguous point where a system can be considered up and running.

The zones restarter is notified of the state of the `milestone/goals` service of each non-global zone that supports it. The zone SMF instance state of each non-global zone matches the state of its `milestone/goals`.

The zones delegated restarter can be instructed to ignore `milestone/goals` to move the zone SMF instance online, based on the success of the `zoneadm boot`

command. If the boot operation fails, the zone SMF instance is placed in maintenance. This switch is controlled by the following SMF property group under the zones delegated restarter service:

```
PG name=config type=application
    config/track-zone-goals = true | false
```

Zone SMF Instance Service States

As part of service management duties, the `svc.zones` command implements a state machine for each of its managed zone SMF instances. The states in this machine are made up of the states described in the [smf\(7\)](#) man page. The following are descriptions of these states:

degraded

The zone SMF instance reached the `online` state, but `zoneadmd` failed multiple times and the daemon is not running.

disabled

The zone is halted and cannot be booted up by the zones restarter.

maintenance

Use the output of the `svcs -vx` command to determine the reason for the `maintenance` state. A zone SMF instance in the `maintenance` state will not clear the `maintenance` state on a disable event. A zone SMF instance can be in the `maintenance` state for the following reasons:

- Failed boot attempt.
- The zone is in a state from which it cannot be booted.
- The zone's `milestone/goals` dependencies cannot be satisfied without administrative action.

offline

The zone SMF instance is enabled, but its SMF dependencies have not been satisfied.

online

The zone is running and has reached `milestone/goals`. For zones that do not support `milestone/goals`, the `online` state will be reached immediately, when the zone begins booting.

The `milestone/goals` provide an unambiguous point where a system can be considered up and running. The dependencies of `milestone/goals` should be configured to align with the mission critical services for the system. The default dependency of `milestone/goals` is:

```
svc:/milestone/multi-user-server:default
```

See [Creating a Goal Service in Developing System Services in Oracle Solaris 11.4](#) for more information about `milestone/goals`.

uninitialized

`svc.zones` has not yet processed the corresponding zone.

SMF Goal Services and Zones

A goal service defines the set of services that must be running for the zone to function as intended. The goal service notifies you if any of these services are not running at system start.

The `milestone/goals` goal service provides a clear, unambiguous, and well-defined point where a system can be considered up and running. The dependencies of `milestone/goals` should be configured to represent the mission critical services for the system. The default dependency of `milestone/goals` is: `svc:/milestone/multi-user-server:default`.

System administrators can use goal services to define the expected set of services that should be running when a system is up and functioning for its intended purpose. Goals services are administered by using the `goals svcadm` subcommand.

Goal services are expected to have all their dependencies satisfiable without intervention. If administrative intervention is required to have a dependency satisfied, the goal service will be placed in the `maintenance` state. Goal services that are in the `maintenance` state will automatically leave that state once their dependencies are satisfied.

In general, the goal service setting should not be used for services that perform actual work.

A service is set to behave as a goal service when the property `general/goal-service=` property is set to `true`.

Goal services should not depend on dynamically enabled services. Having dynamically enabled services will cause a goal service to enter the `maintenance` state until the dependency is enabled by another service. You can use `svcadm` subcommand `goals` to set the dependencies of goal services.

Zones Boot Ordering

Two steps establish zones boot order:

1. Boot order is partially determined by the SMF dependencies of a zone SMF instance.

For example, if ZoneA should be booted before ZoneB, add ZoneA as a dependent of ZoneB. See the `smf-dependency` property in the [zonecfg\(8\)](#) man page.
2. Boot order is further controlled by assigning a boot priority for each zone. After the SMF dependencies are satisfied for a zone, then the zone will boot according to its configured `boot-priority` resource property. Accepted values are `high`, `normal`, and `low`. The default value is `normal`.

See [boot-priority Global Property in Oracle Solaris Zones Configuration Resources](#) and [smf-dependency Resource Type in Oracle Solaris Zones Configuration Resources](#).

Administering the Zones Delegated Restarter

This section provides instructions for administering the zones delegated restarter. For information about the zones delegated restarter, see [Zones Delegated Restarter](#) and the [svc.zones\(8\)](#) man page. For information common to all restarters, see the [smf_restarter\(7\)](#) man page.

- Use the `svcs -R` command to show whether `svc:/system/zones:default` is the restarter for the non-global zone:

```
global$ svcs -R svc:/system/zones:default
STATE      STIME      FMRI
online     12:11:12  svc:/system/zones/zone:zonename
```

- Use the following command to set a property of the zones delegated restarter:

```
global$ pfbash svccfg -s svc:/system/zones:default setprop config/property=value
global$ svcadm refresh svc:/system/zones:default
```

Properties for the zones delegated restarter are as follows:

`config/concurrent-boot-shutdown`

The maximum number of concurrent zones booting up or shutting down.

`config/concurrent-suspend-resume`

The maximum number of concurrent zones performing suspend or resume operations.

For additional information about zones delegated restarter properties, see the [svc.zones\(8\)](#) man page.

Zones Delegated Restarter

Non-global zones are managed by the zones delegated restarter service. The zones delegated restarter runs under the zones service Fault Management Resource Identifier (FMRI), `svc:/system/zones:default`.

This section provides the following information about the zones delegated restarter:

- [How the Zones Delegated Restarter Works](#)
- [Zones Delegated Restarter Properties](#)

How the Zones Delegated Restarter Works

Each SMF service instance is managed by a restarter. The restarter retrieves instance configuration and provides an execution environment. See the [smf_restarter\(7\)](#) man page for information common to all restarters.

A zone SMF instance with the FMRI `svc:/system/zones/zone:zonename` is created by an `install`, `attach`, or `clone zone` operation. The zone SMF instance is removed from the system upon `uninstall` and `detach zone` operations.

Upon startup, the zones delegated restarter creates a zone SMF instance for any non-global zone that requires an SMF instance. If there is an existing zone SMF instance for a zone but it is in a state lesser than the instance `configured`, the restarter will remove the existing SMF instance.

The zones delegated restarter sets up the infrastructure necessary for each zone. The restarter performs the following steps when started:

- Initiates a `zoneadmd` daemon for each zone, restarting it when necessary. There will be a running `zoneadmd` for each zone that is in a state greater than `configured`.
- Runs the `zoneadm boot` command for all zones in the state `installed`, and runs the `zoneadm attach` command for all zones in the state `unavailable`. For more information, see the `zoneadm(8)` man page.

A zones delegated restarter provides the script in the zones service methods. The zones delegated restarter will be running under the existing zones service FMRI.

Zones Delegated Restarter Properties

The zones delegated restarter can limit the number of concurrent zones booting up or shutting down, and suspending or resuming. The limits are set with the `svccfg` and `svcadm` commands by setting the following properties for the zones delegated restarter:

`config/concurrent-boot-shutdown`

The maximum number of concurrent zones booting up or shutting down.

`config/concurrent-suspend-resume`

The maximum number of concurrent zones performing suspend or resume operations.

By default the zones delegated restarter performs tasks for all zones in parallel. There is no limit for concurrent zones operations by the zones restarter if the properties are absent, have no value, or have a value set to zero.

The booting process of a non-global zone is considered completed when the zone SMF instance reaches the `online` state or the `maintenance` state. See [About the Zone SMF Instance Service State](#) for more information about the SMF states for zones.

For more information about the zones delegated restarter, see the `svc.zones(8)` man page.

Reporting the Status of SMF Services in a Non-Global Zone

You can check the status of a non-global zone as you log in to the zone, or you can check it after you log in.

Example 5-2 Checking the Status of SMF Services in a Non-Global Zone at Login

This command displays all services in the zone `my-zone`, including disabled ones, as the administrator logs in.

```
global$ zlogin my-zone svcs -a
```

Example 5-3 Checking the Status of SMF Services From Within a Zone

In this example, the administrator logs in to the zone `my-zone` the zone and then displays all services, including disabled ones.

```
global$ zlogin my-zone
my-zone$ svcs -a
```

Reporting Resource Usage in a Non-Global Zone

The `zonestat` utility reports on the CPU, memory, network, and resource control utilization of the currently running zones. For complete information about the utility, see the [zonestat\(1\)](#) man page.

The `zonestat` network component shows the usage of virtual network (VNIC) resources on PHYS, AGGR, Etherstub, and SIMNET datalinks by zones. You can obtain information about other datalinks, such as bridges and tunnels, by using the networking utilities described in the [dladm\(8\)](#) and [dlstat\(8\)](#) man pages.

All `zonestat` options and resource types can also be invoked within a non-global zone to display statistics for that zone.

```
root@zoneA:~# zonestat -z global -r physical-memory 2
```

Note:

Non-global zone users of `zonestat` are not aware of other zones using system resources. When the `zonestat` utility is used in a non-global zone, the combined resource usage of all other zones, including the global zone, is reported as used by the global zone.

The following examples assume that you logged in to `my-zone` and are assigned zone administration rights for `my-zone`:

- [Displaying the CPU and Memory Utilization of a Non-Global Zone](#)
- [Reporting the Default Processor Set of a Non-Global Zone](#)
- [Reporting the Total and High Utilization of a Non-Global Zone](#)

[Displaying Exclusive-IP Zones' Network Bandwidth From the Global Zone](#) creates a report on several zones from the global zone. For more information about resource usage reports, see the [zonestat\(1\)](#) man page.

Example 5-4 Displaying the CPU and Memory Utilization of a Non-Global Zone

```
my-zone$ zonestat -z global -r physical-memory 5
Collecting data for first interval...
Interval: 1, Duration: 0:00:05
PHYSICAL-MEMORY          SYSTEM MEMORY
mem_default              2046M
                           ZONE  USED  %USED  CAP  %CAP
                           [total] 1020M 49.8%  -   -
                           [system] 782M 38.2%  -   -
                           global 185M 9.06%  -   -

Interval: 2, Duration: 0:00:10
PHYSICAL-MEMORY          SYSTEM MEMORY
mem_default              2046M
                           ZONE  USED  %USED  CAP  %CAP
                           [total] 1020M 49.8%  -   -
```

```
[system] 782M 38.2% - -
global 185M 9.06% - -
...
```

Example 5-5 Reporting the Default Processor Set of a Non-Global Zone

The following command reports on the default `pset` resource pool element of `my-zone` once a second for 1 minute.

```
my-zone$ zonestat -r default-pset 1 1m
Collecting data for first interval...
Interval: 1, Duration: 0:00:01
PROCESSOR_SET          TYPE ONLINE/CPUS      MIN/MAX
pset_default          default-pset      2/2          1/-
                       ZONE USED  PCT  CAP  %CAP  SHRS  %SHR %SHRU
                       [total] 0.02 1.10% - - - - -
                       [system] 0.00 0.19% - - - - -
                       global 0.01 0.77% - - - - -
                       zone1 0.00 0.07% - - - - -
                       zone2 0.00 0.06% - - - - -
```

```
...
Interval: 60, Duration: 0:01:00
PROCESSOR_SET          TYPE ONLINE/CPUS      MIN/MAX
pset_default          default-pset      2/2          1/-
                       ZONE USED  PCT  CAP  %CAP  SHRS  %SHR %SHRU
                       [total] 0.06 3.26% - - - - -
                       [system] 0.00 0.18% - - - - -
                       global 0.05 2.94% - - - - -
                       zone1 0.00 0.06% - - - - -
                       zone2 0.00 0.06% - - - - -
```

Example 5-6 Reporting the Total and High Utilization of a Non-Global Zone

The following command silently monitors at 10-second intervals for 3 minutes, then produces a report on the total and high utilizations of `my-zone`.

```
my-zone$ zonestat -q -R total,high 10s 3m 3m
Report: Total Usage
Start: Fri Aug 26 07:32:22 PDT 2011
End: Fri Aug 26 07:35:22 PDT 2011
Intervals: 18, Duration: 0:03:00
SUMMARY          Cpus/Online: 2/2  PhysMem: 2046M  VirtMem: 3069M
---CPU--- --PhysMem-- --VirtMem-- --PhysNet--
ZONE USED %PART USED %USED USED %USED PBYTE %PUSE
[total] 0.01 0.62% 1020M 49.8% 1305M 42.5% 14 0.00%
[system] 0.00 0.23% 782M 38.2% 1061M 34.5% - -
global 0.00 0.38% 185M 9.06% 208M 6.77% 0 0.00%
test2 0.00 0.00% 52.4M 2.56% 36.6M 1.19% 0 0.00%

Report: High Usage
Start: Fri Aug 26 07:32:22 PDT 2011
End: Fri Aug 26 07:35:22 PDT 2011
Intervals: 18, Duration: 0:03:00
SUMMARY          Cpus/Online: 2/2  PhysMem: 2046M  VirtMem: 3069M
---CPU--- --PhysMem-- --VirtMem-- --PhysNet--
ZONE USED %PART USED %USED USED %USED PBYTE %PUSE
[total] 0.01 0.82% 1020M 49.8% 1305M 42.5% 2063 0.00%
[system] 0.00 0.26% 782M 38.2% 1061M 34.5% - -
global 0.01 0.55% 185M 9.06% 207M 6.77% 0 0.00%
test2 0.00 0.00% 52.4M 2.56% 36.6M 1.19% 0 0.00%
```

Example 5-7 Displaying Exclusive-IP Zones' Network Bandwidth From the Global Zone

The following command shows the network bandwidth utilization for the system's exclusive-IP zones. It shows how much datalink bandwidth in the form of VNICs is in use and the per-zone utilization of each network device.

```
global$ zonestat -r network 1 1
Collecting data for first interval...
Interval: 1, Duration: 0:00:01
```

NETWORK-DEVICE		SPEED		STATE	TYPE			
aggr1		2000mbps		up	AGGR			
	ZONE	TOBYTE	MAXBW	%MAXBW	PRBYTE	%PRBYTE	POBYTE	%POBYTE
	global	1196K	-	-	710K	0.28%	438K	0.18%
net0		1000mbps		up	PHYS			
	ZONE	TOBYTE	MAXBW	%MAXBW	PRBYTE	%PRBYTE	POBYTE	%POBYTE
	[total]	7672K	-	-	6112K	4.89%	1756K	1.40%
	global	5344K	100m*	42.6%	2414K	1.93%	1616K	1.40%
	zoneB	992K	100m	15.8%	1336K	0.76%	140K	0.13%
zoneA	1336K	50m	10.6%	950K	1.07%	0	0.00%	
net1		1000mbps		up	PHYS			
	ZONE	TOBYTE	MAXBW	%MAXBW	PRBYTE	%PRBYTE	POBYTE	%POBYTE
	global	126M	-	-	63M	6.30%	63M	6.30%
etherstub1			n/a		n/a		ETHERSTUB	
	ZONE	TOBYTE	MAXBW	%MAXBW	PRBYTE	%PRBYTE	POBYTE	%POBYTE
	[total]	3920K	-	-	0	-	0	-
	global	1960K	100M*	1.96%	0	-	0	-
	zoneA	1960K	50M	3.92%	0	-	0	-

**Tip:**

Tip: To display the specific VNICs, add the -x option.

Reporting Per-Zone File System Statistics

The following examples show how to report file system activity for non-global zones. For more examples, see the [fsstat\(8\)](#) man page.

Example 5-8 Monitoring Activity in Specific Zones

This command reports on file system activity for specified zones. The separate -z options specify each zone to monitor.

```
global$ pfexec fsstat -z ZoneA -z ZoneB zfs tmpfs
```

new	name	name	attr	attr	lookup	rddir	read	read	write	write	
file	remov	chng	get	set	ops	ops	ops	bytes	ops	bytes	
93	82	6	163K	110	507K	148	69.7K	67.9M	4.62K	13.7M	zfs:ZoneA
248	237	158	188K	101	612K	283	70.6K	68.6M	4.71K	15.2M	zfs:ZoneB
12.0K	1.90K	10.1K	35.4K	12	60.3K	4	25.7K	29.8M	36.6K	31.0M	tmpfs:ZoneA
12.0K	1.90K	10.1K	35.6K	14	60.2K	2	28.4K	32.1M	36.5K	30.9M	tmpfs:ZoneB

**Tip:**

To report on all zones, including the global zone, use the `-Z` option (uppercase) instead of the `-z zonename` option.

Example 5-9 Displaying Per-Zone `fstype` Statistics for All Zones

This example shows how to report aggregated file system activity for all zones, including the global zone. The administrator requests reports on the file system types `tmpfs` and `zfs` for each zone running on the system.

```
global$ pfexec fsstat -A -Z zfs tmpfs
new name name attr attr lookup rddir read read write write
file remov chng get set ops ops ops bytes ops bytes
360K 1.79K 20.2K 4.20M 1.02M 25.0M 145K 5.42M 2.00G 1.07M 8.10G zfs
359K 1.48K 20.1K 4.04M 1.02M 24.5M 144K 5.31M 1.88G 1.06M 8.08G zfs:global
 93 82 6 74.8K 107 250K 144 54.8K 60.5M 4.61K 13.7M zfs:ZoneA
 248 237 158 90.2K 101 336K 283 53.0K 58.3M 4.71K 15.2M zfs:ZoneB
60.0K 41.9K 17.7K 410K 515 216K 426 1022K 1.02G 343K 330M tmpfs
49.4K 38.1K 11.0K 366K 489 172K 420 968K 979M 283K 273M tmpfs:global
5.28K 1.90K 3.36K 21.9K 12 21.7K 4 25.7K 29.8M 29.9K 28.3M tmpfs:ZoneA
5.25K 1.90K 3.34K 22.1K 14 21.6K 2 28.4K 32.1M 29.8K 28.2M tmpfs:ZoneB
```

Using the Fair Share Scheduler on a System With Zones Installed

This section shows how to use the FSS for zones. For more information, see the [prctl\(1\)](#) man page.

How to Temporarily Set FSS Shares in the Global Zone

The global zone is given one share by default. Perform this procedure to temporarily change the default allocation.

- 1. Become a zone administrator.**

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

- 2. Assign shares to the global zone.**

```
global$ prctl -n zone.cpu-shares -v N -r -i zone global
```

-v value

Specifies the new value, in this case, the new number of CPU shares.

-i zone idtype

Specifies the zone name or the zone ID.

- 3. Verify the number of shares assigned to the global zone.**

```
global$ prctl -n zone.cpu-shares -i zone global
zone: 0: global
NAME PRIVILEGE VALUE FLAG ACTION
RECIPIENT
zone.cpu-shares
```

```
usage          1
privileged     1      -  none
system        65.5K   max  none
```

Example 5-10 Dynamically Changing the zone.cpu-shares Value in a Zone

```
$ prctl -n zone.cpu-shares -r -v 2 -i zone ZoneA
```

6

About Non-Global Zone Installation and Packages

This chapter discusses non-global zone features that differ from global zone features.

This chapter covers the following topics:

- [Packages and Zones on an Oracle Solaris 11.4 System](#)
- [Zone Contents After Creation](#)
- [Zone Information After Installation](#)
- [Boot Environments and Zones](#)

Packages and Zones on an Oracle Solaris 11.4 System

This section supplements [Chapter 3, Installing and Updating Software Packages in *Updating Systems and Adding Software in Oracle Solaris 11.4*](#).

Package management in `solaris` non-global zones is similar to management in global zones, with the following differences:

- Only a subset of the Oracle Solaris packages from the global zone are completely replicated when a non-global zone is installed. For example, many Oracle Solaris kernel packages are not needed in a non-global zone. All non-global zones share the same kernel from the global zone.
- If a package is installed in the global zone, then the non-global zone can install the package from the global zone's system-repository service and does not have to use the network to install that package. If that package has not been installed in the global zone, then the zone will need to use the zones-proxy service to access the publishers to install the package over the network, using the global zone. See [Configuring Proxies to the Package Repository for Non-Global Zones](#).
- The root (/) file system for a non-global zone can be administered from the global zone by using the Oracle Solaris packaging tools. The packaging tools in the non-global zone can administer co-packaged (bundled), standalone (unbundled), or third-party products.
- When a zone package variant is specified, the various components within a package are specifically tagged to only be installed in either a global zone or a non-global zone. A given package can contain a file that is tagged so that it will not be installed into a non-global zone.

For more information, see [Working with Non-Global Zones in *Updating Systems and Adding Software in Oracle Solaris 11.4*](#) and [Automatically Installing Oracle Solaris 11.4 Systems](#).

 **Note:**

While certain package operations are performed, a zone is temporarily locked to other operations of this type. The system might also confirm a requested operation with the administrator before proceeding. When updating the global zone on a system with non-global zones, the system might appear to display package download information twice for the zones. However, the packages are downloaded only once.

Configuring Proxies to the Package Repository for Non-Global Zones

You have two options when connecting to the package repository from a zone. You can create persistent proxies, or specify the proxies during package install.

- You set persistent proxies in an image by using the `--proxy` option as described in [Chapter 5, Configuring Installed Images in *Updating Systems and Adding Software in Oracle Solaris 11.4*](#).
- To specify proxies during install, you modify the `system-repository` service properties to use the same proxies as the `http_proxy` and `https_proxy` environment variables. See the `pkg(1)` man page.

Access to repositories configured in the global zone is provided to non-global zones using the `system-repository` service. Any updates to proxies for origins in the global zone are automatically made to the `system-repository` configuration. Using this method, no modifications are required to the `system-repository` SMF service, as the following example illustrates.

Example 6-1 Configuring the Proxy in the Global Zone

```
$ pfexec pkg set-publisher --proxy http://www-proxy -g http://pkg-server pub
```

No port specification is required unless the proxy accepts connections on a port other than 80. If zones are on the system, the `system-repository` service is restarted, and the proxy is used to provide access to `pkg-server`.

Overriding `system-repository` Proxies by Using `https_proxy` and `http_proxy`

Proxies should be set in an image, and only the `system-repository` service proxy should be set. The `https_proxy` and `http_proxy` environment variables should be set in the environment where you run the `pkg` command.

You can also configure the proxies used by the `system-repository` SMF service, overriding any proxies configured on publishers in the global zone. The `system-repository` proxies can be set by using the `config/http_proxy` or `config/https_proxy` SMF properties.

For more information, see [Updating Systems and Adding Software in Oracle Solaris 11.4](#) and the following procedure, [How to Set `https_proxy` and `http_proxy` To Override Global Zone Proxies](#).

This procedure sets proxies in the `system-repository` service on an internal subnet that does not have a direct connection to the IPS publisher repository. This configuration overrides any proxies that the `pkg` command configured in the global

zone. Non-global zones communicate with the `system-repository` over HTTP. The `system-repository` then accesses the publishers using the protocol for that repository as configured in the global zone.

This configuration allows the `solaris` non-global zones to contact the publisher set in the global zone as well. Recursive `pkg` operations into the `solaris` zones will succeed.

How to Set `https_proxy` and `http_proxy` To Override Global Zone Proxies

The following steps show how to use the `http_proxy` and `https_proxy` environment variables and SMF service properties to allow the global zone and non-global zones to access the IPS repositories.

Note that these environment variables override any proxy configuration set on the origin, unless the user is running the `pkg` command from a non-global zone to connect to the Universal Resource Identifier (URI) for a system publisher. In that case, the command goes through the `system-repository`.

A host name that can be resolved can also be used.

1. Set the proxy in the shell for the global zone.

Setting the proxy enables `pkg` commands to reach the publisher through the proxy server. This affects `pkg` operations that use an `https` or `http` URL and do not go through the `system-repository` for the global zone. For more information about the `pkg` command, see the [`pkg\(1\)`](#) man page.

```
$ export http_proxy=http://192.0.2.0
$ export https_proxy=http://192.0.2.0
```

2. Enable the `solaris` zones on the system to use the configured system publishers that are directly accessible from the global zone.

```
$ svccfg -s system-repository:default setprop config/http_proxy = http://192.0.2.0
$ svccfg -s system-repository:default setprop config/https_proxy = http://192.0.2.0
```

3. Make the change take effect in the live SMF repository.

```
$ svcadm refresh system-repository
```

4. Confirm that the setting is in effect.

```
$ svcprop -p config/http_proxy system-repository
$ svcprop -p config/https_proxy system-repository
```

See Also

See Also

you can update zones in parallel instead of serially. A parallel update provides a significant improvement in the time required to update all the zones on a system. For details and a configuration example, see [Updating Multiple Non-Global Zones Concurrently in *Updating Systems and Adding Software in Oracle Solaris 11.4*](#).

Zone Contents After Creation

This section applies to initial non-global zone construction, and not to the cloning of existing zones.

A zone is installed with the packages that are specified by the manifest passed to the `zoneadm install -m` command. If no manifest is provided, the `zone_default.xml` manifest uses `pkg:/group/system/solaris-small-server`.

A new zone has the default `solaris` configuration and logs, which are only modified by the system configuration profiles that are passed to the `zoneadm install -s` command, and the networking information specified in any `zonecfg add net` entries.

The system repository, the zone's configured publishers, and packages kept in sync with the global zone are discussed in [Packages and Zones on an Oracle Solaris 11.4 System](#).

The files needed for the zone's root file system are installed by the system under the zone's root path.

A successfully installed zone is ready for booting and initial login.

Data from the following are not referenced or copied when a zone is installed:

- Non-installed packages
- Data on CDs and DVDs
- Network installation images

In addition, the following types of information that can be present in the global zone are not copied into a zone that is being installed:

- New or changed users in the `/etc/passwd` file
- New or changed groups in the `/etc/group` file
- Configurations for networking services such as DHCP address assignment
- Customizations for networking services such as `sendmail`
- Configurations for network services such as naming services
- New or changed `crontab`, printer, and mail files
- System log, message, and accounting files
- Audit logs

The resources that are specified in the configuration file are added when the zone state transitions from `installed` to `ready`. A unique zone ID is assigned by the system. File systems are mounted, network interfaces are set up, and devices are configured. Transitioning into the `ready` state prepares the virtual platform to begin running user processes.

In the `ready` state, the `zsched` and `zoneadmd` processes are started to manage the virtual platform.

- The `zsched` process is a system scheduling process similar to the `sched` process. It is used to track kernel resources associated with the zone.
- The `zoneadmd` process is the zones administration daemon.

A zone in the `ready` state does not have any user processes executing in the zone. At least one process must be executing in a zone for the zone to be in the `running` state. See the [init\(8\)](#) man page for more information.

Zone Information After Installation

The `zoneadmd` daemon manages zones, the `zsched` process tracks the kernel use by the zone, and by using the `sysconfig` command, the administrator sets up the zone application environment.

`zoneadmd` Zones Administration Daemon

The zones administration daemon, `zoneadmd`, is the primary process for managing the zone's virtual platform. The daemon is also responsible for managing zone booting and shutting down. There is one `zoneadmd` process running for each active (ready, running, or shutting down) zone on the system.

The `zoneadmd` daemon sets up the zone as specified in the zone configuration. This process includes the following actions:

- Allocating the zone ID and starting the `zsched` system process
- Setting zone-wide resource controls
- Preparing the zone's devices as specified in the zone configuration
- Setting up network interfaces
- Mounting loopback and conventional file systems
- Instantiating and initializing the zone console device

If the `zoneadmd` daemon is not already running, it is automatically started by the `zoneadm` command. Thus, if the daemon is not running for any reason, any invocation of the `zoneadm` command to administer the zone restarts the `zoneadmd` daemon.

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

`zsched` Zone Scheduler Process

An active zone is a zone that is in the `ready`, `running`, or `shutting_down` state. Every active zone has an associated kernel process, `zsched`. Kernel threads doing work on behalf of the zone are owned by the `zsched` process. The `zsched` process enables the zones subsystem to keep track of per-zone kernel threads.

Boot Environments and Zones

This section describes zone boot environments (BEs), shared zones, alternate roots for zones, and orphaned zones.

`beadm` Command in Non-Global Zones

The `beadm` command is used to manage boot environments, called BEs. The `beadm` command performs tasks on a boot environment structure including a root dataset and all the datasets nested under that root dataset.

The `beadm` command is supported inside a non-global zone. The `beadm` command does not display zones information. Use the `zoneadm` command from the global zone to view changes

in the zones in your boot environment. For example, use the `zoneadm list` command to display a list of all current zones on the system. For more information, see the [beadm\(8\)](#) man page.

Boot Environments in Global and Non-Global Zones

Both global zones and non-global zones contain boot environments. Each boot environment in a non-global zone is associated with a parent boot environment in the global zone. Each global zone boot environment is associated with one or more zone boot environments (ZBE) in each installed `solaris` zone. When the zone is detached, the references to the global zone remain. This enables the zone to be reattached to the same global zone while maintaining associations between global and non-global boot environments.

If a global zone boot environment is inactive, the related non-global zone boot environment is unbootable. However, if you boot into that parent boot environment in the global zone, the related boot environment in the non-global zone becomes bootable.

When you attach or migrate a zone, you can specify the ZBE to attach by name using the `-z` option or use one of several `-x` options to enable the system to select the appropriate ZBE. See the [zones\(7\)](#) man page for information about these options.

-x deny-zbe-clone Option

This option denies a zone boot environment (ZBE) clone, updates the ZBE, and mounts it as active boot environment.

By default, if either of the following are true, the selected zone boot environment is cloned and the clone is attached. The original zone boot environment continues to exist.

- The selected zone boot environment is associated with another global zone boot environment.
- The selected zone boot environment is an orphaned boot environment that is not associated with any global zone boot environment.

If the `-x deny-zbe-clone` option is used during attach or migration of a `solaris` zone, the chosen zone boot environment is updated in place and mounted as the active boot environment without cloning.



Note:

You can only use the `-x deny-zbe-clone` option on an inactive zone boot environment. To mark a boot environment inactive, see [Chapter 4, Administering Boot Environments in *Creating and Administering Oracle Solaris 11.4 Boot Environments*](#).

-x destroy-orphan-zbes Option

During `attach` or `migrate` operations, this option destroys all zone boot environments that are not associated with any existing global zone. The current orphan boot environment you attach will be destroyed after it is cloned and updated.

-x force-zbe-clone Option

This option clones the selected zone boot environment (ZBE) and attaches the clone ZBE as the active boot environment. The original ZBE remains.

When attaching a zone, by default, the zone boot environment that is associated with the currently active global zone boot environment is selected. This option overrides this behavior and selects a different global zone boot environment.

Shared Zone State

The state of a zone is shared across all boot environments (BEs) on a host system. If the state of a zone is changed, it affects all BEs, not just the BE that is currently booted. It also affects all states, which includes changing the zone state from `installed` to `configured`. If a zone is uninstalled, it is uninstalled from all BEs. The only way to recover the uninstalled zone is from existing backup data.

Specifying an Alternate Root

Use the `zoneadm` command with the `-R root` option to specify an alternate root (boot environment). This option can only be used with the `list` and `mark` subcommands. See the [`zoneadm\(8\)`](#) man page for more information.

About Orphaned Zone Boot Environments and Clones

When migrated to different hosts, non-global zones can accumulate zone boot environments that are not associated with an existing global zone. A non-global zone boot environment that is not associated with a global zone is known as an **orphaned boot environment**.

Orphaned zone boot environments are of use only if you intend to reattach the zone on the original source system. If there is no intent to return the zone to a global zone to which it was previously attached, you can destroy the orphaned zone boot environment.

If an orphaned boot environment is selected to attach to a zone, it is cloned. This cloned boot environment will be updated and re-parented to the active global boot environment. The original orphaned boot environment will continue to exist by default. You can specify options to handle the boot environments differently.

Identifying Orphaned Boot Environments With the `beadm` Command

Orphaned boot environments are indicated in `beadm list` output by an `O` flag in the `Active` column. In the following example output, the `solaris-0`, `solaris-1`, and `solaris-2` boot environments are orphaned.

```
$ beadm list
  BE          Active Mountpoint Space   Policy Created
  --          -
solaris-0 !RO -          3.40M static 2014-01-17 15:04
solaris-1 !RO -          3.85M static 2014-02-09 18:11
solaris-2 !RO -          39.0K static 2014-02-10 04:54
solaris-3 !R  -          2.19G static 2014-02-12 16:56
solaris-4 !R  -          3.0K  static 2014-02-19 19:18
solaris-5 NR  /          2.43G static 2014-03-20 04:38
```

Boot environments that are shown as orphaned can be destroyed with the `beadm destroy` command. To destroy all orphaned boot environments, include the `-O` option.

For more information about boot environments, see [Creating and Administering Oracle Solaris 11.4 Boot Environments](#) and the `beadm(8)` man page.

zoneadm Options for Orphaned Zone Boot Environments and Clones

The following `-x` options are used to manage zone boot environments (ZBEs) during `zoneadm attach` of a zone. The same options are used similarly with the `zoneadm migrate` command and apply when an attach is performed on the target system during the migration.

- `destroy-orphan-zbes`
- `force-zbe-clone`
- `deny-zbe-clone`



Note:

The `force-zbe-clone` and the `deny-zbe-clone` are mutually exclusive options. The options `attach-last-booted-zbe` and `destroy-orphan-zbes` can be used with either the `force-zbe-clone` option or the `deny-zbe-clone` option.

7

About Non-Global Zone Administration

This chapter covers the following topics:

- [Security Measures for a System With Non-Global Zones](#)
- [Device Use in Non-Global Zones](#)
- [Resource Management With Non-Global Zones](#)
- [Monitoring a System With Non-Global Zones](#)
- [File Systems and Non-Global Zones](#)
- [Networking in Exclusive-IP Non-Global Zones](#)
- [Networking in Shared-IP Non-Global Zones](#)
- [Migration and Transformation of Non-Global Zones](#)
- [Backup Strategies for a System With Non-Global Zones](#)
- [Commands Used on a System With 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*](#).

Security Measures for a System With Non-Global Zones

Oracle Solaris provides similar security measures to global and non-global zones.

IPsec and IKE protect the network, rights and auditing prevent unauthorized use of resources, and immutable zones prevent zone modification.

- For information about rights profiles for zones, see [Using Rights Profiles to Install and Manage Zones](#).
- For information about privileges in zones, see [Privileges in a Non-Global Zone](#).
- For information about preventing zone modification, see [Configuring and Administering Immutable Zones](#).
- For information about auditing zones, refer to:
 - [Auditing on a System With Oracle Solaris Zones in *Managing Auditing in Oracle Solaris 11.4*](#)
 - [Configuring the Audit Service in Zones in *Managing Auditing in Oracle Solaris 11.4*](#)
- For information about IPsec and IKE, and Packet Filter, see [Exclusive-IP Zones Traffic, Traffic Security, and IPMP Configuration](#) and [Shared-IP Zones Traffic, Traffic Security, and IPMP Configuration](#).

Privileges in a Non-Global Zone

Zone processes are restricted to a subset of privileges to prevent a zone from affecting other zones, including the global zone. To display the privileges available in a zone, type the following from the appropriate zone:

```
global$ ppriv -l zonename
```

```
zonename> ppriv -l
```

Not all privileges that are installed by default are necessary. However, zones must keep the following privileges:

```
file_read
file_write
net_access
proc_exec
proc_fork
sys_linkdir
sys_net_config
sys_res_config
sys_smb
sys_suser_compat
```

You can add privileges to a zone's default privileges. For example, see [Adding DTrace Privileges to a Non-Global Zone](#). However, the following privileges are reserved for the global zone and cannot be added to a zone:

```
dtrace_kernel
proc_zone
sys_config
sys_devices
sys_dl_config
sys_linkdir
sys_ip_config
sys iptun_config
sys_mount
```

Caution:

Applications that rely on privileges that are reserved for the global zone cannot be run in a non-global zone.

For further information about privileges, review the following:

- [Process Rights Management in *Securing Users and Processes in Oracle Solaris 11.4*](#)
- [privileges\(7\)](#) man page
- [Zone Administration Utilities in Trusted Extensions in *Trusted Extensions Configuration and Administration*](#)

Device Use in Non-Global Zones

The set of devices available within a zone is restricted to prevent a process in one zone from interfering with processes running in other zones. For example, a process in a zone cannot modify kernel memory or modify the contents of the root disk. Thus, by default, only certain pseudo-devices are considered safe for use in a zone. Additional devices can be made available within specific zones by using the `zonecfg` utility.

▲ Caution:

Applications that depend on devices that do not exist in a non-global zone, such as `/dev/kmem`, cannot be run in a non-global zone.

`/dev` and the `/devices` Namespace

The `devfs` file system described in the [devfs\(4FS\)](#) man page is used by the Oracle Solaris system to manage `/devices`. Each element in this namespace represents the physical path to a hardware device, pseudo-device, or nexus device. The namespace is a reflection of the device tree. As such, the file system is populated by a hierarchy of directories and device special files.

Devices are grouped according to the relative `/dev` hierarchy. For example, all of the devices under `/dev` in the global zone are grouped as global zone devices. For a non-global zone, the devices are grouped in a `/dev` directory under the zone's root path. Each group is a mounted `/dev` file system instance that is mounted under the `/dev` directory. Thus, the global zone devices are mounted under `/dev`, while the devices for a non-global zone named `my-zone` are mounted under `/my-zone/root/dev`.

The `/dev` file hierarchy is managed by the `dev` file system.

▲ Caution:

Subsystems that rely on `/devices` path names are not able to run in non-global zones. The subsystems must be updated to use `/dev` path names.

▲ Caution:

If a non-global zone has a device resource with a match that includes devices within `/dev/zvol`, namespace conflicts might occur within the non-global zone. For more information, see the [dev\(4FS\)](#) man page.

Exclusive-Use Devices and Zones

You might have devices that you want to assign to specific zones. Allowing unprivileged users to access block devices could permit those devices to be used to cause system panic, bus

resets, or other adverse effects. Before making such assignments, consider the following issues:

- Before assigning a SCSI tape device to a specific zone, consult the [sgen\(4D\)](#) man page.
- Placing a physical device into more than one zone can create a covert channel between zones. Global zone applications that use such a device risk the possibility of compromised data or data corruption by a non-global zone.

Device Driver Administration in Zones

In a non-global zone, you can use the `modinfo` command described in the [modinfo\(8\)](#) man page to examine the list of loaded kernel modules.

Most operations concerning kernel, device, and platform management will not work inside a non-global zone because modifying platform hardware configurations violates the zone security model. These operations include the following:

- Adding and removing drivers
- Explicitly loading and unloading kernel modules
- Initiating dynamic reconfiguration (DR) operations
- Using facilities that affect the state of the physical platform

Device Utilities and Non-Global Zones

Some utilities cannot work in non-global zones, some can be modified for use within a zone, and the use of some utilities has security implications.

Device Utilities That Do Not Work in Non-Global Zones

The following utilities do not work in a zone because they rely on devices that are not normally available:

- `add_drv` (see the [add_drv\(8\)](#) man page)
- `disks` (see the [disks\(8\)](#) man page)
- `prtconf` (see the [prtconf\(8\)](#) man page)
- `prtdiag` (see the [prtdiag\(8\)](#) man page)
- `rem_drv` (see the [rem_drv\(8\)](#) man page)

SPARC: `eeprom` in Non-Global Zones

The `eeprom` utility can be used in a zone to view settings but not to change settings. For more information, see the [eeprom\(8\)](#) and [openprom\(4D\)](#) man pages.

Device Utilities With Security Implications

After security considerations have been evaluated and `allowed-raw-io` is enabled, the following utilities can be used in a zone. Review [Device Use in Non-Global Zones](#) and [Privileges in a Non-Global Zone](#) for restrictions and security concerns.

- `cdrecord` (see the `cdrecord(1)` man page).
- `cdrw` (see the `cdrw(1)` man page).
- `rmformat` (see the `rmformat(1)` man page).

Resource Management With Non-Global Zones

This section covers resource controls, the Fair Share Scheduler (FSS), and extended accounting.

Resource Controls Used in Non-Global Zones

For additional information about using a resource management feature in a zone, also refer to the chapter that describes the capability in *Administering Resource Management in Oracle Solaris 11.4*.

Any of the resource controls and attributes described in the resource management chapters can be set in the `/etc/project` file, NIS map, or LDAP directory service in all zones. The settings for a given zone affect only that zone. A project running autonomously in different zones can have controls set individually in each zone. For example, Project A in the global zone can be set `project.cpu-shares=10` while Project A in a non-global zone can be set `project.cpu-shares=5`. You could have several instances of `rcapd` running on the system, with each instance operating only on its zone.

The resource controls and attributes used in a zone to control projects, tasks, and processes within that zone are subject to the additional requirements regarding pools and the zone-wide resource controls.

A non-global zone can be associated with one resource pool, although the pool need not be exclusively assigned to a particular zone. Multiple non-global zones can share the resources of one pool. Processes in the global zone, however, can be bound by a sufficiently privileged process to any pool. The resource controller `poolld` only runs in the global zone, where there is more than one pool for it to operate on. The `poolstat` utility run in a non-global zone displays only information about the pool associated with the zone. The `pooladm` command run without arguments in a non-global zone displays only information about the pool associated with the zone.

Zone-wide resource controls do not take effect when they are set in the `project` file. A zone-wide resource control is set through the `zonecfg` utility.

Fair Share Scheduler on a System With Zones Installed

This section describes how to use the fair share scheduler (FSS) with zones to balance CPU use.

Share Balance Between Zones

You can use `zone.cpu-shares` to assign FSS shares in the global zone and in non-global zones. If FSS is the default scheduler on your system and shares are not assigned, each zone is given one share by default. If you have one non-global zone on your system and you give this zone two shares through `zone.cpu-shares`, that defines the proportion of CPU which the non-global zone will receive in relation to the global zone. The ratio of CPU between the two zones is 2:1.

Extended Accounting on a System With Zones Installed

The extended accounting subsystem collects and reports information for the entire system (including non-global zones) when run in the global zone. The global administrator can also determine resource consumption on a per-zone basis.

The extended accounting subsystem permits different accounting settings and files on a per-zone basis for process-based and task-based accounting. The `exacct` records can be tagged with the zone name `EXD PROC ZONENAME` for processes, and the zone name `EXD TASK ZONENAME` for tasks. Accounting records are written to the global zone's accounting files as well as the per-zone accounting files. The `EXD TASK HOSTNAME`, `EXD PROC HOSTNAME`, and `EXD HOSTNAME` records contain the `uname -n` value for the zone in which the process or task executed instead of the global zone's node name.

Monitoring a System With Non-Global Zones

In general, non-global zones and their processes are visible to the global zone but cannot see each others' processes. As on systems without zones, you can limit what is visible remotely, create core files, set time, run DTrace, and view statistics.

Global Zone Visibility and Access

The global zone acts as both the default zone for the system and as a zone for system-wide administrative control. There are administrative issues associated with this dual role. Since applications within the zone have access to processes and other system objects in other zones, the effect of administrative actions can be wider than expected. For example, service shutdown scripts often use `pkill` to signal processes of a given name to exit. When such a script is run from the global zone, all such processes in the system will be signaled, regardless of zone.

The system-wide scope is often needed. For example, to monitor system-wide resource usage, you must view process statistics for the whole system. A view of just global zone activity would miss relevant information from other zones in the system that might be sharing some or all of the system resources. Such a view is particularly important when system resources such as CPU are not strictly partitioned using resource management facilities.

Thus, processes in the global zone can observe processes and other objects in non-global zones. This allows such processes to have system-wide observability. The ability to control or send signals to processes in other zones is restricted by the privilege `proc_zone`. The privilege is similar to `proc_owner` because the privilege allows processes to override the restrictions placed on unprivileged processes. In this case, the restriction is that unprivileged processes in the global zone cannot signal or control processes in other zones. This is true even when the user IDs of the processes match or the acting process has the `proc_owner` privilege. The `proc_zone` privilege can be removed from otherwise privileged processes to restrict actions to the global zone.

For information about matching processes by using a `zoneidlist`, see the [pgrep\(1\)](#) and [pkill\(1\)](#) man pages.

Process ID Visibility in Zones

Only processes in the same zone will be visible through system call interfaces that take process IDs, such as the `kill` and `priocntl` commands. For information, see the [kill\(1\)](#) and the [priocntl\(1\)](#) man pages.

System Observability in Zones

When the `ps` command is executed in the global zone, user and group names are resolved using the global zone's name services. Processes running in a non-global zone on the system display user and group names that match the global zone's name services. These global zone names might be different than the names configured in name services in the non-global zones.

The `ps` command has the following modifications for zones:

- The `-o` option is used to specify output format. This option allows you to print the zone ID of a process or the name of the zone in which the process is running.
- The `-z zonelist` option is used to list only processes in the specified zones. Zones can be specified either by zone name or by zone ID. This option is only useful when the command is executed in the global zone.
- The `-Z` option is used to print the name of the zone associated with the process. The name is printed under the column heading `ZONE`.

For more information, see the [ps\(1\)](#) man page.

A `-z zonename` option has been added to the following Oracle Solaris utilities. You can use this option to filter the information to include only the zone or zones specified.

- `ipcs -z zonename` – See the [ipcs\(1\)](#) man page
- `pgrep -z zonename` – See the [pgrep\(1\)](#) man page
- `ptree -z zonename` – See the [proc\(1\)](#) man page
- `prstat -z zonename` – See the [prstat\(8\)](#) man page

For the full list of changes made to commands, see [Commands With Modifications for Zones](#).

Reporting Active Zone Statistics With the `zonestat` Utility

To use the `zonestat` utility, see the [zonestat\(1\)](#) man page and [Reporting Resource Usage in a Non-Global Zone](#).

The `zonestat` utility reports on the CPU, memory, and resource control utilization of the currently running zones. The `zonestat` utility prints a series of reports at specified intervals. Optionally, the utility can print one or more summary reports.

The `zonestat` utility also reports on network bandwidth utilization in exclusive-IP zones. An exclusive-IP zone has its own IP-related state and one or more dedicated datalinks.

When executed from within a non-global zone, only processor sets visible to that zone are reported. The non-global zone output will include all of the memory resources, and the limits resource.

The `zonestat` service in the global zone must be online to use the `zonestat` service in the non-global zones. The `zonestat` service in each non-global zone reads system configuration and utilization data from the `zonestat` service in the global zone.

The `zonestatd` system daemon is started during system boot. The daemon monitors the utilization of system resources by zones, as well as zone and system configuration information such as `psrset` processor sets, pool processor sets, and resource control settings. There are no configurable components.

Monitoring Non-Global Zones With the `fsstat` Utility

The `fsstat` utility collects and prints `kstats` per zone, including aggregations. By default, the utility reports an aggregate of all running zones. A per-`fstype` `kstat` is produced for each zone. The global zone `kstat` reports its exclusive activity. The global zone can see the `kstats` of all zones on the system. Non-global zones only see the `kstats` associated with the zone in which the utility is run. A non-global zone cannot monitor file system activity in other zones.

For more information, see the `fsstat(8)` man page and [Reporting Per-Zone File System Statistics](#).

Running DTrace in a Non-Global Zone

DTrace programs that require only the `dtrace_proc` and `dtrace_user` privileges can be run in a non-global zone. To add these privileges to the set of privileges available in the non-global zone, set the zone's `limitpriv` property, as shown in [Adding DTrace Privileges to a Non-Global Zone](#).

DTrace supports the providers `fasttrap` and `pid` through `dtrace_proc`. The providers supported through `dtrace_user` are `profile` and `syscall`. DTrace providers and actions operate only within the zone.

Core Files in Zones

The `coreadm` command is used to specify the name and location of core files produced by abnormally terminating processes. Core file paths that include the `zonename` of the zone in which the process executed can be produced by specifying the `%z` variable. The path name is relative to a zone's root directory.

For more information, see the `coreadm(8)` and `core(5)` man pages.

Non-Global Zone Node Name

The node name is the local source for the system name. The node name must be unique, such as the zone name. The zone node name can be set by the zone administrator. For example, set the zone name in the global zone:

```
$ hostname kzone1
```

To display the hostname, type the `hostname` command without an argument. For example:

```
$ hostname  
kzone1
```

File Systems and Non-Global Zones

This section provides information about file system issues on an Oracle Solaris system with zones installed. Issues include using the zone as an NFS server or client, mounting and traversing mounted file systems, and file system restrictions that are specific to zones.

Each zone has its own section of the file system hierarchy, rooted at a directory known as the zone `root`. Processes in the zone can access files only in the file system hierarchy that is located under the zone `root`. The `chroot` utility can be used in a zone, but only to restrict the process to a root path within the zone. For more information about `chroot`, see the [chroot\(8\)](#) man page.

Running an NFS Server in a Zone

The NFS server package `svc:/network/nfs/server:default` must be installed in the zone to create NFS shares in a zone.

The `sys_share` privilege can be prohibited in the zone configuration to prevent NFS sharing within a zone. See [Privileges in a Non-Global Zone](#).

Restrictions and limitations include the following:

- Cross-zone LOFS mounts cannot be shared from zones.
- File systems mounted within zones cannot be shared from the global zone.
- NFS over Remote Direct Memory Access (RDMA) is not supported in zones.
- Oracle Sun Cluster HA for NFS (HANFS) failover is not supported in zones.

For more information, see [Introduction to Oracle Solaris 11.4 Network Services](#).

Mounting File Systems in Zones

When file systems are mounted from within a zone, the `nodevices` option applies. For example, if a zone is granted access to a block device (`/dev/dsk/c0t0d0s7`) and a raw device (`/dev/rdisk/c0t0d0s7`) corresponding to a UFS file system, the file system is automatically mounted `nodevices` when mounted from within a zone. This rule does not apply to mounts specified through a `zonecfg` configuration.



Note:

Allowing file system mounts other than the default might enable the zone administrator to compromise the system.

Options for mounting file systems in non-global zones are described in the following table. Any file system type not listed in the table can be specified in the configuration if it has a mount binary in `/usr/lib/ fstype /mount`.

To mount file system types other than HSFs and NFS from inside the non-global zone, also add the file system type to the configuration by using the `zonecfg fs-allowed` property.

File System	Mounting Options in a Non-Global Zone
AutoFS	Cannot be mounted using <code>zonecfg</code> . Can be mounted from within the zone.
CacheFS	Cannot be used in a non-global zone.
FDFS	Can be mounted using <code>zonecfg</code> . Can be mounted from within the zone.
HSFS	Can be mounted using <code>zonecfg</code> . Can be mounted from within the zone.
LOFS	Can be mounted using <code>zonecfg</code> . Can be mounted from within the zone.
MNTFS	Cannot be mounted using <code>zonecfg</code> . Can be mounted from within the zone.
NFS	Cannot be mounted using <code>zonecfg</code> . V2, V3, and V4, which are the versions currently supported in zones, can be mounted from within the zone.
PCFS	Can be mounted using <code>zonecfg</code> . Can be mounted from within the zone.
PROCFS	Cannot be mounted using <code>zonecfg</code> . Can be mounted from within the zone.
TMPFS	Can be mounted using <code>zonecfg</code> . Can be mounted from within the zone.
UDFS	Can be mounted using <code>zonecfg</code> . Can be mounted from within the zone.
UFS	<p>Can be mounted using <code>zonecfg</code>. Can be mounted from within the zone.</p> <p>The <code>quota</code> command, which is documented in the quota(8) man page, cannot be used to retrieve quota information for UFS file systems that are added with the <code>fs</code> resource of the <code>zonecfg</code> command.</p> <p>The <code>system/file-system/ufs</code> package must be installed in the global zone if the <code>fs</code> resource is used. To use UFS file systems in a non-global zone through the <code>zonecfg</code> command, the package must be installed into the zone after installation or through the AI manifest script.</p> <pre>global\$ pfexec pkg -R /system/zones/zonename/root install file-system/ufs</pre>
VxFS	Can be mounted using <code>zonecfg</code> . Can be mounted from within the zone.
ZFS	Can be mounted using the <code>zonecfg</code> dataset and <code>fs</code> resource types.

For mounting procedures, see [Installing a Non-Global Zone](#) and [Adding File Systems for Zone Use](#).

For more information, see [How to Create and Deploy a Non-Global Zone](#) and the [mount\(8\)](#) man page.

Unmounting File Systems in Zones

The ability to unmount a file system will depend on who performed the initial mount. If a file system is specified as part of the zone's configuration using the `zonecfg` command, then the global zone owns this mount and the non-global zone administrator cannot unmount the file system. If the file system is mounted from within the non-global zone, for example, by specifying the mount in the zone's `/etc/vfstab` file, then the non-global zone administrator can unmount the file system.

Non-Global Zones as NFS Clients

Zones can be NFS clients. NFS version 2, version 3, and version 4 protocols are supported. For information about these NFS versions, see [Features of the NFS Service in *Managing Network File Systems in Oracle Solaris 11.4*](#).

The default version is NFS version 4. You can enable other NFS versions on a client by using one of the following methods:

- **Use the `sharectl` command to set properties** – See [How to Select Different Versions of NFS on a Client in *Managing Network File Systems in Oracle Solaris 11.4*](#) and the `sharectl(8)` man page.
- **Manually create a version mount** – This method overrides the `sharectl` setting. See [How to Select Different Versions of NFS on a Server in *Managing Network File Systems in Oracle Solaris 11.4*](#).

Traversing File Systems

A zone's file system namespace is a subset of the namespace accessible from the global zone. Unprivileged processes in the global zone are prevented from traversing a non-global zone's file system hierarchy through the following means:

- Specifying that the zone root's parent directory is owned, readable, writable, and executable by root only
- Restricting access to directories exported by `/proc`

Note that attempting to access AutoFS nodes mounted for another zone will fail. The global administrator must not have auto maps that descend into other zones.

Security Restrictions and File System Behavior

Mounting certain file systems from within a zone presents a security risk. Non-default file systems exhibit special behavior when mounted in a zone. The list of modified file systems follows.

AutoFS

AutoFS is a client-side service that automatically mounts the appropriate file system. AutoFS mounts established within a zone are local to that zone and cannot be accessed from other zones, including the global zone. The mounts are removed when the zone is halted or rebooted. For more information about AutoFS, see [How Autofs Works in *Managing Network File Systems in Oracle Solaris 11.4*](#).

Each zone runs its own copy of `automountd`. The auto maps and timeouts are controlled by the zone administrator.

An AutoFS mount that is created in the kernel when another mount is triggered cannot be removed by using the regular `umount` interface. These mounts are unmounted as a group during zone shutdown.

MNTFS

MNTFS is a virtual file system that provides read-only access to the table of mounted file systems for the local system. The set of file systems visible by using `mnttab` from within a non-global zone is the set of file systems mounted in the zone, plus an entry for root (`/`). All mounts in the system are visible from the global zone's `/etc/mnttab` table. For more

information about MNTFS, see [Mounting File Systems in Managing Network File Systems in Oracle Solaris 11.4](#).

NFS

NFS mounts that are established within a zone are local to that zone and cannot be accessed from other zones, including the global zone. The mounts are removed when the zone is halted or rebooted.

From within a zone, NFS mounts behave as though mounted with the `nodevices` option.

The `nfsstat` command output only pertains to the zone in which the command is run. For more information, see [nfsstat\(8\)](#).

PROCFS

The `/proc` file system, or PROCFS, provides process visibility and access restrictions as well as information about the zone association of processes. Only processes in the same zone are visible through `/proc`.

Processes in the global zone can observe processes and other objects in non-global zones.

From within a zone, `procfs` mounts behave as though mounted with the `nodevices` option. For more information about `procfs`, see the [proc\(5\)](#) man page.

LOFS

The scope of what can be mounted through LOFS is limited to the portion of the file system that is visible to the zone. Hence, there are no restrictions on LOFS mounts in a zone.

UFS, UDFS, PCFS, and other storage-based file systems

When using the `zonecfg` command to configure storage-based file systems that have an `fsck` binary, such as UFS, the zone administrator must specify a `raw` parameter. The parameter indicates the raw (character) device, such as `/dev/rdisk/c0t0d0s7`. The `zoneadmd` daemon automatically runs the `fsck` command in preen mode (`fsck -p`), which checks and fixes the file system non-interactively, before it mounts the file system. If the `fsck` fails, `zoneadmd` cannot bring the zone to the ready state. The path specified by `raw` cannot be a relative path.

It is an error to specify a device to `fsck` for a file system that does not provide an `fsck` binary in `/usr/lib/fs/fstype/fsck`. It is also an error if you do not specify a device to `fsck` if an `fsck` binary exists for that file system.

For more information, see [zoneadmd Zones Administration Daemon](#) and the [fsck\(8\)](#) man page.

ZFS

In addition to the default dataset described in [File Systems Mounted in Zones in Oracle Solaris Zones Configuration Resources](#), you can add a ZFS dataset to a non-global zone by using the `zonecfg` command with the `add dataset` resource. The dataset is visible and mounted in the non-global zone, and also visible in the global zone. The zone administrator can create and destroy file systems within that dataset, and modify the properties of the dataset.

The `zoned` attribute of `zfs` indicates whether a dataset has been added to a non-global zone.

```
$ pfexec zfs get zoned dataset
NAME          PROPERTY  VALUE  SOURCE
dataset       zoned     on     local
```

Each dataset that is delegated to a non-global zone through a dataset resource is aliased. The dataset layout is not visible within the zone. Each aliased dataset appears in the zone as if it were a pool. The default alias for a dataset is the last component in the dataset name. For example, if the default alias is used for the delegated dataset `tank/sales`, the zone will see a virtual ZFS pool named `sales`. The alias can be customized to be a different value by setting the `alias` property within the dataset resource.

A dataset named `rpool` exists within each non-global zone's `zonepath` dataset. For all non-global zones, this zone `rpool` dataset is aliased as `rpool`.

```
zonename$ zfs list -o name,zoned,mounted,mountpoint
NAME                ZONED  MOUNTED  MOUNTPOINT
rpool                on      no       /rpool
rpool/ROOT           on      no       legacy
rpool/ROOT/solaris  on      yes      /
rpool/export         on      no       /export
rpool/export/home   on      no       /export/home
```

Dataset aliases are subject to the same name restrictions as ZFS pools. These restrictions are documented in the [zpool\(8\)](#) man page.

If you want to share a global zone dataset, you can add an LOFS-mounted ZFS file system by using the `zonecfg` command with the `add fs` subcommand. An administrator with the appropriate rights is responsible for setting and controlling the properties of the dataset. For more information about ZFS, see [Chapter 10, Oracle Solaris ZFS Advanced Topics in *Managing ZFS File Systems in Oracle Solaris 11.4*](#).

Restriction on Using `mknod` in a Zone

You cannot use the `mknod` command to make a special file in a non-global zone.

Restriction on Accessing a Non-Global Zone From the Global Zone

After a non-global zone is installed, the zone must never be accessed directly from the global zone by any commands other than system backup utilities. Moreover, a non-global zone can no longer be considered secure after it has been exposed to an unknown environment. An example would be a zone placed on a publicly accessible network, where it would be possible for the zone to be compromised and the contents of its file systems altered. If any compromise could have occurred, the global administrator should treat the zone as untrusted.

Any command that accepts an alternative root by using the `-R` or `-b` options (or the equivalent) must *not* be used when the following are true:

- The command is executed in the global zone.
- The alternative root refers to any path within a non-global zone, whether the path is relative to the current running system's global zone or the global zone in an alternative root.

An example is the `pkgadd -R root-path` command when run from the global zone with a non-global zone root path.

- Commands that use `-R` with an alternative root path include `auditreduce`, `metaroot`, `pkg`, and `syseventadm`.
- Commands that use `-b` with an alternative root path include `add_drv` and `useradd`.

Networking in Exclusive-IP Non-Global Zones

Network in exclusive-IP zones is similar to networking in the global zone. Some administration of exclusive-IP networking is done in the zone, and some is done from the global zone.

Overview of Networking in Exclusive-IP Zones

An exclusive-IP zone has its own IP-related state. The zone is assigned its own set of datalinks when the zone is configured.

Packets are transmitted on the physical link. Then, devices like Ethernet switches or IP routers can forward the packets toward their destination, which might be a different zone on the same system as the sender.

For virtual links, the packet is first sent to a virtual switch. If the destination link is over the same device, such as a VNIC on the same physical link or etherstub, the packet will go directly to the destination VNIC. Otherwise, the packet will go out the physical link underlying the VNIC.

For information about features that can be used in an exclusive-IP non-global zone, see [Exclusive-IP Non-Global Zones in Oracle Solaris Zones Configuration Resources](#).

Exclusive-IP Zone Partitioning

Exclusive-IP zones have separate TCP/IP stacks, so the separation reaches down to the datalink layer. One or more datalink names, which can be a NIC or a VLAN on a NIC, are assigned to an exclusive-IP zone by the global administrator. The zone administrator can configure IP on those datalinks with the same flexibility and options as in the global zone.

Exclusive-IP Datalink Interfaces

A datalink name must be assigned exclusively to a single zone.

The `dladm show-link` command displays datalinks assigned to running zones, similar to the following example:

```
example-114{jdoe}1: dladm show-linkLINK                CLASS    MTU    STATE
OVER
vsw0                phys     1500   up     --
net0                phys     1500   up     --
netg2               phys     1500   up     --
netg1               phys     1500   up     --
netg3               phys     1500   up     --
zoneA/net0          vnic     1500   up     net0
zoneB/net0          vnic     1500   up     net0
aggr1               aggr     1500   up     net2 net3
vnic0               vnic     1500   up     net1
zoneA/vnic0         vnic     1500   up     net1
vnic1               vnic     1500   up     net1
zoneB/vnic1         vnic     1500   up     net1
vnic3               vnic     1500   up     aggr1
vnic4               vnic     1500   up     aggr1
zoneB/vnic4         vnic     1500   up     aggr1
```


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

Exclusive-IP Zones Traffic, Traffic Security, and IPMP Configuration

- **Traffic Between Zones** – There is no internal loopback of IP packets between exclusive-IP zones. All packets are sent down to the datalink. Typically, this means that the packets are sent out on a network interface. Then, devices like Ethernet switches or IP routers can forward the packets toward their destination, which might be a different zone on the same system as the sender.
- **IPsec and IKE** – You have the same IPsec and IKE functionality that you have in the global zone in an exclusive-IP zone. PF is also configured the same way in exclusive-IP zones and the global zone. See [IPsec Reference in *Securing the Network in Oracle Solaris 11.4*](#).
- **Packet Filter Firewall** – You have the same Packet Filter (PF) functionality that you have in the global zone in an exclusive-IP zone. PF is also configured the same way in exclusive-IP zones and the global zone. See [Chapter 5, *Configuring the Firewall in Oracle Solaris in Securing the Network in Oracle Solaris 11.4*](#).
- **IP Network Multipathing (IPMP)** – The datalink configuration is done in the global zone. First, multiple datalink interfaces are assigned to a zone by using the `zonecfg` command. The multiple datalink interfaces must be attached to the same IP subnet. IPMP can then be configured from within the exclusive-IP zone by the zone administrator.

IPMP is used for physical interface failure detection and transparent network access failover for a system with multiple interfaces on the same IP link. IPMP also provides load spreading of packets for systems with multiple interfaces.

Networking in Shared-IP Non-Global Zones

Shared-IP zones have separate bindings, or connections, and can run their own server daemons. These daemons can listen on the same port numbers without any conflict. The IP stack resolves conflicts by considering the IP addresses for incoming connections. The IP addresses identify the zone.

Shared-IP Zone Partitioning

Shared-IP is not the default, but this type is supported.

The IP stack in a system supporting zones implements the separation of network traffic between zones. Applications that receive IP traffic can only receive traffic sent to the same zone.

Each logical interface on the system belongs to a specific zone, the global zone by default. Logical network interfaces assigned to zones through the `zonecfg` utility are used to communicate over the network. Each stream and connection belongs to the zone of the process that opened it.

Bindings between upper-layer streams and logical interfaces are restricted. A stream can only establish bindings to logical interfaces in the same zone. Likewise, packets from a logical interface can only be passed to upper-layer streams in the same zone as the logical interface.

Each zone has its own set of bindings. Each zone can be running the same application listening on the same port number without binds failing because the address is already in use. Each zone can run its own version of various networking service such as the following:

- Internet services daemon with a full configuration file (see the [inetd\(8\)](#) man page)
- `sendmail` (see the [sendmail\(8\)](#) man page)
- `apache`

Zones other than the global zone have restricted access to the network. The standard TCP and UDP socket interfaces are available, but `SOCK_RAW` socket interfaces are restricted to Internet Control Message Protocol (ICMP). ICMP is necessary for detecting and reporting network error conditions or using the `ping` command.

Shared-IP Network Interfaces

Each non-global zone that requires network connectivity has one or more dedicated IP addresses. These addresses are associated with logical network interfaces that can be placed in a zone. Zone network interfaces configured by `zonecfg` will automatically be set up and placed in the zone when it is booted. The `ipadm` command can be used to add or remove logical interfaces when the zone is running. Only the global administrator or a user granted the appropriate authorizations can modify the interface configuration and the network routes.

Within a non-global zone, only that zone's interfaces are visible to the `ipadm` command.

For more information, see the [ipadm\(8\)](#) and [if_tcp\(4P\)](#) man pages.

Shared-IP Zones Traffic, Traffic Security, and IPMP Configuration

- Traffic Between Zones – A shared-IP zone can reach any given IP destination if there is a usable route for that destination in its routing table. To view the routing table, use the `netstat -r` command from within the zone. The IP forwarding rules are the same for IP destinations in other zones or on other systems.
- IPsec and IKE – IPsec relies on the Internet Key Exchange (IKE) protocol to manage keys. If you are configuring IPsec in a shared-IP zone, configure IKE in the global zone and use the source address that corresponds to the non-global zone that you are configuring. See [IPsec Reference in *Securing the Network in Oracle Solaris 11.4*](#).
- Packet Filter Firewall – PF can be enabled in non-global zones by turning on loopback filtering as described in [Chapter 5, Configuring the Firewall in Oracle Solaris in *Securing the Network in Oracle Solaris 11.4*](#).
- IP Network Multipathing (IPMP) – You configure IPMP in the global zone. Then, you extend the functionality to non-global zones. The functionality is extended by assigning one of the IPMP interface's data addresses to the zone. In a given non-global zone, only the interfaces associated with the zone are visible through the `ipadm` command.

For further information, review the following:

- [How to Extend IP Network Multipathing Functionality to Shared-IP Non-Global Zones](#)
- [Chapter 2, About IPMP Administration in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*](#)
- [How to Create and Deploy a Non-Global Zone](#)

Migration and Transformation of Non-Global Zones

Migration means the moving of a zone to a new location. *Transformation* means converting the zone to an archive that can be deployed widely. Oracle Solaris provides the `zonep2vchk` tool to assist in zone transformations.

About Cold Migration of `solaris` Zones

A **zone migration** is the process of transferring an existing zone configuration and data from one host system to another host system. Zone migrations can only be performed for zones on shared storage. Zone migration can be cold, warm, or live. However, because `solaris` zones depend on the host's kernel, the zones cannot be running during migration to a new host. Therefore, warm migration and live migration are not supported for a `solaris` zone.

Cold migration occurs when the zone is not running on the source host while the zone is migrated. During cold migration, the zone is detached from storage, and a small amount of data is moved to the new host. When the zone is attached on the new host, it attaches the shared storage and the zone data is accessed using the same path. Cold migrations take approximately the same amount of time as a zone detach followed by an attach.

For additional information about shared storage, see [Oracle Solaris Zones on Shared Storage](#). See [About Zone Migration in Introduction to Oracle Solaris Zones](#) for more information about zone migration and zone migration types.

Zone Migration Configuration Requirements

Before migrating, you need to consider configuration and host requirements, and package updates.

- **Zone state** – The zone's state must be `installed` when you begin the migration. After migration, the state will still be `installed` on the new host.
- **Shared storage** – If the zone is not on shared storage, you can use Unified Archives to move the zone, similar to a zone transformation. See [About Zone Transformations](#).
- **User authorizations** – Migrations must be performed by a user or role that has the Zone Migration and Zone Configuration profiles. You can authorize specific non-root users to perform migrations. See [Authorizing Non-Root Users to Perform Non-Global Zone Migrations](#) for more information.

Zone resources – A `solaris` zone configuration must meet the following resource requirements to be migrated:

device Resource Type

Any `device` resources must be specified with storage URI. The `match` property cannot be set.

fs, dataset Resource Types

The `fs` and `dataset` resources are not allowed for cold migration because they refer to local file systems or local devices.

npiv:over-hba Resource Type Property

The `npiv:over-hba` property can only be set if the zone exists on the target system. This setting is bonded to a physical controller number, which is likely not consistent between the source and target systems. However, if the zone configuration exists on the target system before the migration, it is assumed that you have set up the systems so that the controller numbers are consistent and migration should be possible.

rootzpool, zpool Resource Types

The `rootzpool` and `zpool` resources must be specified with shared storage resources using `iscsi` and `lu` URIs.

`zoneadm migrate` Command

Use the `zoneadm migrate` command to migrate a zone. See the command syntax for a `solaris` zone in the [zoneadm\(8\)](#) man page.

Zone Migration and Various Host Configurations

Observe the following guidelines concerning the source host and target host of a zone migration:

- If you are migrating a zone to an identical system and all storage references use a storage URI that is accessible by both hosts, the migrated configuration should be compatible without changes. If the target system is not identical, you must modify the configuration on the target system after migration to suit the new system.
- If the zone storage is local, you cannot use the `zoneadm migrate` command. You can instead do one of the following:
 - Remove local storage devices from the zone configuration if they are not needed for booting, and then use `zoneadm migrate`.
 - Convert the storage to shared as described in [How to Move a Zone to a Shared Storage Configuration](#), and then use `zoneadm migrate`.
 - Archive the zone and redeploy on the new system instead of migrating as described in [Archiving and Moving Non-Global Zones That Are Not Using Shared Storage](#).
- If you configure the zone on the target host before migration, the target host's version of the zone configuration is used to boot the zone.
- If you do not configure the zone on the target host before migration, the zone configuration is exported from the source host and imported on the target host. The user performing the migration must have the Zone Configuration rights profile and `solaris.zone.config` authorization to create zone configurations on the target host. See [Authorizing Non-Root Users to Perform Non-Global Zone Migrations](#) for more information.

Zone Migration and Package Updates

`solaris` zones always have their storage mapped and zpools imported, even in an `attached` state. This is necessary so that packages in the zone can be updated when the global zone is updated. When a `solaris` zone is attached on the target system during migration, the packages in the zone are verified against the global zone and the

attach fails if the package levels are different, unless you specify to attempt to update the zone with the `-u` or `-U` update option.

A migration using a `-u` or `-U` update option attempts to update the packages in the `solaris` zone to match the global zone. During that attempt a new **ZBE** might be created.

If no `-x attach-*-zbe` option is given to `zoneadm migrate`, the attach on the destination system selects the most recently mounted ZBE to be cloned. The cloned ZBE is then attached.

About Zone Transformations

Zone transformation or *zone conversion* is the process of creating an archive of an Oracle Solaris global zone or non-global zone and deploying it.

You can perform the following types of zone transformations:

- **Global zone to non-global zone.** Called a **physical to virtual** or **P2V** conversion.
- **Non-global zone to global zone.** Called a **virtual to physical** conversion.
- **Non-global zone to non-global zone.** Called a **virtual to virtual** or **V2V** conversion. This transformation is useful for migrating a zone that is not using shared storage and thus cannot be migrated by using the `zoneadm migrate` command.

Oracle Solaris supports several virtualization technologies. For example, both logical domains and zones are virtual Oracle Solaris instances. Transformation of a logical domain to a zone can be considered a V2V conversion. However, because a logical domain is running a global zone, it is also a global to non-global conversion, which is a P2V conversion. For this reason, the model of zone transformation on Oracle Solaris 11 is usually discussed in terms of global versus non-global, instead of physical versus virtual.

In the Oracle Solaris 11.4 release, you can use only Unified Archives to transform to and from zones. See [Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4](#) for more information about Unified Archives.

Zone transformations are implemented by archiving the Oracle Solaris instance that you want to convert and deploying the archive into the new zone or system.

On host systems running Oracle Solaris 11 releases prior to Oracle Solaris 11.2, you must use legacy archives for zone conversion, as documented in those earlier releases.

Zone Transformations Using Oracle Solaris Unified Archives

Unified Archive files contain both zone configuration and zone data. This means that on the target system a zone can be configured and installed from the archive. You can use the `zonecfg` command to configure and the `zoneadm` command to install new zones directly from a Unified Archive file. See [Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4](#) for a full description of Unified Archives, including usage for system and zone cloning and recovery.

The overall process for zone transformation is as follows:

1. Ensure that both the source host and the target host where the new zone will be located meet requirements. See [Zone Transformation Requirements for Source and Target Hosts](#) for more information.
2. If transforming a global zone to a non-global zone, use the `zonep2vchk` tool to identify any issues that might cause problems in the transformed zone. See [Using the](#)

[zonep2vchk Tool to Prepare for Global to Non-Global Transformations](#) for more information.

3. Create an archive of the global zone by using the `archiveadm` command.
4. Create the target zone configuration by using the `zonecfg` command.
5. Install the zone from the archive by using the `zoneadm` command, or install a system from the archive by using the `installadm` command.

For more information, see the [zonep2vchk\(8\)](#), [archiveadm\(8\)](#), [zfs\(8\)](#), [zonecfg\(8\)](#), and [zoneadm\(8\)](#), and [zones\(7\)](#) man pages.

Zone Transformation Requirements for Source and Target Hosts

Transformation of `solaris` zones has the following requirements:

- **Architecture** – The source and target systems must use the same instruction set architecture (ISA). For example, they must both be SPARC based systems, or must both be x86 based systems.
- **Oracle Solaris versions** – The global zone on the target system must be running an Oracle Solaris release that is equal to or higher than that on the original source host.
 - Before you transform a global zone to a non-global zone, verify that the software running in the global zone is compatible with non-global zones. The [zonep2vchk\(8\)](#) tool evaluates a global zone's configuration before transformation to a non-global zone.
 - Before you transform a non-global zone to a global zone or kernel zone, you must ensure that the Oracle Solaris operating system versions of the source and target hosts are compatible. You do not need to check for additional compatibility. Any software running in the non-global zone can also run in a global zone or kernel zone, provided that the host systems are compatible.
- **Oracle Solaris packages** – To ensure that the zone will run properly, the target system must have the same or later versions of the required operating system packages as those installed on the original source host. For example, if the source host is running a Support Repository Update (SRU), then the target must also be running that SRU or a later SRU.

Other packages, such as those for third-party products, can be different.

Zone Transformations From Systems That Contain Non-Global Zones

When you transform a global zone to a non-global zone, any existing `solaris` zones or kernel zones that are within that global zone will not be usable after the zone is transformed. Only global zones and kernel zones can contain other zones.

Tip:

To retain the `solaris` zones or kernel zones before you transform the global zone, create separate recovery archives for each individual zone. Then you can redeploy the zones from those archives on a new system. For example, you could first create archives for the non-global zones. Then, you could archive the global zone.

For more information, see [Chapter 1, Oracle Solaris System Recovery and Cloning \(Overview\)](#) in *Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4*.

Using the `zonep2vchk` Tool to Prepare for Global to Non-Global Transformations

The `zonep2vchk` tool evaluates a global zone's configuration before transformation to a non-global zone. The primary documentation for the tool is the [`zonep2vchk\(8\)`](#) man page.

This section provides the following information:

- [About the `zonep2vchk` Tool](#)
- [Checks Performed by the `zonep2vchk` Tool](#)
- [Issues Highlighted by the `zonep2vchk` Tool](#)
- [Zone Configuration Template for a Transformed Zone](#)

About the `zonep2vchk` Tool

The `zonep2vchk` tool evaluates a global zone, or physical instance, for potential issues that can occur when it is converted to a non-global zone, or virtual instance. The `zonep2vchk` tool can be run with an effective user `id` of `0`.

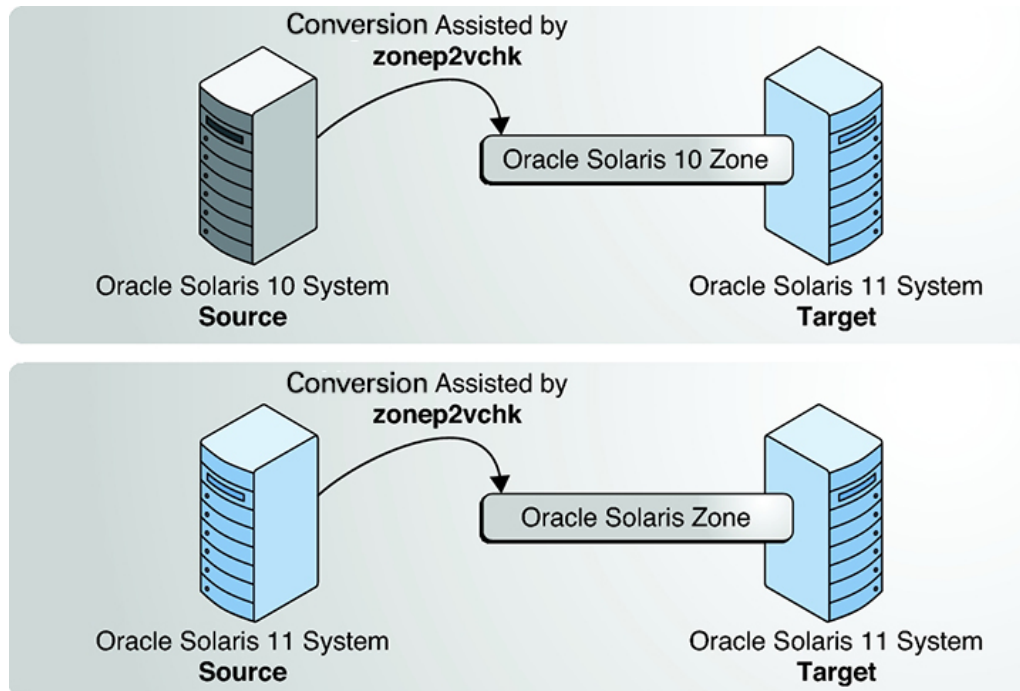
The `zonep2vchk` tool does the following:

- Identifies problem areas in the source system's configuration
- Minimizes the manual reconfiguration effort required
- Supports conversion of Oracle Solaris global zones into non-global zones on the same Oracle Solaris releases
- Supports complex network configurations in the original system image, including multiple IP interfaces, IP multipathing, and VLANs

Use the `zonep2vchk` tool to assist in the transformation of an Oracle Solaris global zone to a non-global zone. The tool evaluates the system to flag any issues that might complicate or prevent transformation to a `solaris` brand zone, and makes suggestions for actions to take to enable conversion. The tool can also provide a template for the new `solaris` zone configuration.

The following figure shows using `zonep2vchk` to aid physical system transformation to a zone on an Oracle Solaris system.

`zonep2vchk` Evaluation Scenarios



Checks Performed by the `zonep2vchk` Tool

You can use `zonep2vchk` options to perform several types of checking:

- **Basic analysis** checks for Oracle Solaris features in use that might be impacted by global to non-global transformation. This is the default if you specify no options. Use the `-b` option for basic checking when combined with other options.
- **Static analysis** inspects binaries for system and library calls that might not function in a zone. Use the lowercase `-s` option to specify paths for binaries to analyze, or use the uppercase `-S` option to specify the path to a file that lists the files and directories to analyze.
- **Runtime analysis** inspects the currently executing applications for operations that might not function in a zone. Use the `-r` option with a specified duration in hours, minutes, or seconds, or the `-x` option to analyze until you send a signal interrupt such as by typing `Ctrl-C`.

For more information, see [How to Check the Source System With `zonep2vchk`](#) and the `zonep2vchk(8)` man page.

Issues Highlighted by the `zonep2vchk` Tool

The `zonep2vchk` tool produces the following main categories of information when you run the various checks:

- Issues that can be addressed with a specific zone configuration or with configuration changes in the global zone
- Identification of functions that cannot work inside a zone

For example, if an application sets the system clock, that ability can be enabled in a `solaris` zone by adding the appropriate privilege to the zone. However, an application that accesses kernel memory cannot run in a `solaris` zone. The output distinguishes between these two classes of issues.

By default, the `zonep2vchk` tool prints messages in human readable form. To print messages in machine parsable form, use the `-P` option. For more information, see the [zonep2vchk\(8\)](#) man page.

Zone Configuration Template for a Transformed Zone

You can use the `zonep2vchk -c` command that to create a template for use with the `zonecfg` command. The template configures some of the global zone's resources appropriately for the non-global zone.

You can run the analysis checks and make any necessary changes in the global zone before creating the template. You can create the template by running the command and directing the output to a file which can then be used as input to the `zonecfg` command to create a starting point for the zone configuration. For example:

```
global$ pfbash zonep2vchk -c > myzone.config
global$ zonecfg -z myzone -f myzone.config
```

Backup Strategies for a System With Non-Global Zones

You can perform backups in individual non-global zones, or back up the entire system from the global zone:

Backup Methods for a System With Non-Global Zones

From within a zone you can back up databases or applications only, or the entire zone. You can also back up a non-global zone from the global zone. Backups are easily restored.

Backing Up Loopback File System Directories

Back up and restore loopback file systems (`lofs`) from the global zone only, to avoid multiple copies. Do not back up the loopback file systems of non-global zones.

If you back up and restore `read/write` loopback file systems from within a non-global zone, these file systems are then also writable from the global zone and from any other zones in which they are `read/write` mounted.

Backing Up Your System From the Global Zone

You might choose to perform your backups from the global zone in the following cases:

- You want to back up the configurations of your non-global zones as well as the application data.
- Your primary concern is the ability to recover from a disaster. If you need to restore everything or almost everything on your system, including the root file systems of your zones and their configuration data as well as the data in your global zone, backups should take place in the global zone.
- You have commercial network backup software.

 **Note:**

Your network backup software should be configured to skip all inherited `lofs` file systems if possible. The backup should be performed when the zone and its applications have quiesced the data to be backed up.

Backing Up Individual Non-Global Zones on Your System

You might decide to perform backups from within a non-global zone in the following cases.

- The non-global zone administrator needs the ability to recover from less serious failures or to restore application or user data specific to a zone.
- You use the backup software of a particular application or service running in a zone. It might be difficult to execute the backup software from the global zone because application environments, such as directory path and installed software, would be different between the global zone and the non-global zone.

If the application can perform a snapshot on its own backup schedule in each non-global zone and store those backups in a writable directory exported from the global zone, the global zone administrator can pick up those individual backups as part of the backup strategy from the global zone.

Creating Oracle Solaris Unified Archive Backups

You can use the `zonecfg` and `zoneadm` commands to configure and to install new zones directly from a Unified Archive file. Unified Archive files contain both zone configuration and zone data. On the destination system, a zone can be configured and installed from the archive.

For more information, see [Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4](#) for a full description of Unified Archives, including usage for system and zone cloning and recovery.

Determining What to Back Up in Non-Global Zones

You can back up everything in the non-global zone, or you can back up the application data only.

Backing Up Application Data Only

If application data is kept in a particular part of the file system, you might decide to perform regular backups of this data only. The zone's root file system might not have to be backed up as often because it changes less frequently.

You will have to determine where the application places its files. Locations where files can be stored include the following directories:

- User home directories
- `/etc` for configuration data files
- `/var`

Assuming the application administrator knows where the data is stored, it might be possible to create a system in which a per-zone writable directory is made available to each zone. Each zone can then store its own backups, and the global administrator or user granted the appropriate authorizations can make this location one of the places on the system to back up.

General Database Backup Operations

If the database application data is not under its own directory, the following rules apply:

- Ensure that the databases are in a consistent state first.
Databases must be quiesced because they have internal buffers to flush to disk. Make sure that the databases in non-global zones have come down before starting the backup from the global zone.
- Within each zone, use file system capabilities to make a snapshot of the data, then back up the snapshots directly from the global zone.
This process will minimize elapsed time for the backup window and remove the need for backup clients/modules in all of the zones.

Tape Backups

Each non-global zone can take a snapshot of its private file systems when it is convenient for that zone and the application has been briefly quiesced. Later, the global zone can back up each of the snapshots and put them on tape after the application is back in service.

The snapshot method has the following advantages:

- Fewer tape devices are needed.
- There is no need for coordination between the non-global zones.
- There is no need to assign devices directly to zones, which improves security.
- Generally, this method keeps system management in the global zone, which is preferred.

About Restoring Non-Global Zones

In the case of a restore where the backups were performed from the global zone, the global administrator or a user granted the appropriate authorizations can reinstall the affected zones and then restore that zone's files. This assumes the following:

- The zone being restored has the same configuration as it did when the backup was done.
- The global zone has not been updated between the time when the backup was done and the time when the zone is restored.

Otherwise, the restore could overwrite some files that should be merged by hand.



Note:

If all file systems in the global zone are lost, restoring everything in the global zone restores the non-global zones as well, as long as the respective root file systems of the non-global zones were included in the backup.

Backing Up and Recreating Zones

This section provides procedures to back up files in zones and the zone configuration file, and to use a saved configuration file to recreate the zone.

Note:

On a system with zones on shared storage, archive and deploy each global or non-global zone separately. See [How to Archive a System That Contains Zones on Shared Storage in Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4](#).

Example 7-1 Using Unified Archives to Back Up a System With Zones

The following example creates a recovery archive of the zone `my-zone` to `/net/server/my-zone-archive.uar`.

```
host1$ pfbash archiveadm create -r -z my-zone /net/server/my-zone-archive.uar
```

Example 7-2 Printing a Copy of a Zone Configuration

The administrator creates a backup file of the non-global zone configuration at first login after responding to the `sysidtool` questions. The administrator plans to use the backup later to create the zone again.

```
global$ pfbash zonecfg -z my-zone export > /net/server/my-zone-configtemplate.txt
```

Example 7-3 Recreating a Non-Global Zone

The administrator follows these steps to recreate `my-zone` from the files in [Printing a Copy of a Zone Configuration](#) and [Using Unified Archives to Back Up a System With Zones](#).

1. Uses the configuration file to configure the zone.

```
host2$ pfbash zonecfg -z my-zone -f /net/server/my-zone-configtemplate.txt
```

2. Uses the archive to install the zone.

```
host2$ zoneadm -z my-zone install -a /net/server/my-zone-archive.uar
```

Commands Used on a System With Non-Global Zones

This section provides descriptions and man page links to commands that are used with zones.

Commands for Zone Administration and Monitoring

The following commands provide the primary administrative interface to Oracle Solaris Zones. The commands are listed by man page section.

 **Note:**

The `zoneadmd` daemon is the primary process for managing the zone's virtual platform. The daemon does not constitute a programming interface. For more information, see the [zoneadmd\(8\)](#) man page.

getzoneid(3C)

Used to map between zone ID and name

zcons(4D)

Zone console device driver

zlogin(1)

Used to log in to a non-global zone

zoneadm(8)

Used to administer zones on a system

zonecfg(8)

Used to set up a zone configuration

zonename(1)

Prints the name of the current zone

zones(7)

Provides description of zones facility

zonestat(1)

Used to observe zone resource usage.

Commands for Resource Capping

The following commands support the resource capping daemon. The commands are listed by man page section.

rcapadm(8)

Configures the resource capping daemon, displays the current status of the resource capping daemon if it has been configured, and enables or disables resource capping

rcapd(8)

The resource capping daemon.

rcapstat(1)

Monitors the resource utilization of capped projects.

Commands With Modifications for Zones

The following commands are modified for use on an Oracle Solaris system with zones installed. These commands have options that are specific to zones or present information differently. The commands are listed by man page section.

audit_syslog(7)

Added `in zonename` field that is used if the `zonename` audit policy is set.

auditconfig(8)

Added `zonename` token.

auditreduce(8)

Added `-z zone-name` option. Added ability to get an audit log of a zone.

cmn_err(9F)

Added zone parameter.

core(5)

Added `n_type: NT_ZONENAME`. This entry contains a string that describes the name of the zone in which the process was running.

coreadm(8)

Added variable `%z` to identify the zone in which process executed.

ddi_cred(9F)

Added `crgetzoneid()`, which returns the zone ID from the user credential pointed to by `cr`.

df(8)

Added `-Z` option to display mounts in all visible zones. This option has no effect in a non-global zone.

dladm(8)

Added `-Z` option to `show` subcommands, which adds a zone column to the default command output. The zone column indicates the zone to which the resource is currently assigned.

dlstat(8)

Added `-Z` option to `show` subcommands, which adds a zone column to the default command output. The zone column indicates the zone to which the resource is currently assigned.

fsstat(8)

Added `-z` option to report on file system activity per zone. Multiple `-z` options can be used to monitor activity in selected zones. The option has no effect if only used to monitor `mountpoints` and not `fstypes`.

Added `-A` option to report aggregate file system activity for the specified `fstypes` across all zones. This is the default behavior if neither `-z` or the `-Z` option is used. The `-A` option has no effect if only used to monitor `mountpoints` and not `fstypes`.

When used with either the `-z` or the `-Z` option, the `-A` option displays the aggregate for the specified `fstypes` across all zones on a separate line.

Added `-Z` option to report file system activity in all zones on the system. This option has no effect if used with `-z` option. The option has no effect if only used to monitor `mountpoints` and not `fstypes`.

getloadavg(3C)

If the caller is in a non-global zone and the pools facility is enabled, the behavior is equivalent to calling with a `psetid` of `PS_MYID`.

getpriority(3C)

Added zone IDs to target processes that can be specified. Added zone ID to `EINVAL` error description.

if_tcp(4P)

Added zone `ioctl()` calls.

iostat(8)

If executed in a non-global zone in which the pools facility is enabled, information is provided only for those processors that are in the processor set of the pool to which the zone is bound.

ipcrm(1)

Added `-z zone` option. This option is only useful when the command is executed in the global zone.

ipcs(1)

Added `-z zone` option. This option is only useful when the command is executed in the global zone.

kstat(8)

If executed in the global zone, `kstats` are displayed for all zones. If executed in a non-global zone, only `kstats` with a matching `zoneid` are displayed.

mpstat(8)

If executed in a non-global zone in which the pools facility is enabled, command only displays lines for the processors that are in the processor set of the pool to which the zone is bound.

ndd(8)

When used in the global zone, displays information for all zones. `ndd` on the TCP/IP modules in an exclusive-IP zone only displays information for that zone.

netstat(8)

Displays information for the current zone only.

nfsstat(8)

Displays statistics for the current zone only.

p_online(2)

If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.

pgrep(1)

Added `-z zoneidlist` option. This option is only useful when the command is executed in the global zone.

pkginfo(5)

Now provides optional parameters and an environment variable in support of zones.

poolbind(8)

Added `zoneid` list. Also see [Resource Pools Used in Zones in Administering Resource Management in Oracle Solaris 11.4](#) for information about using zones with resource pools.

ppriv(1)

Added the expression `zone` for use with the `-l` option to list all privileges available in the current zone. Also use the option `-v` after `zone` to obtain verbose output.

priocntl(1)

Zone ID can be used in `idlist` and `-i idtype` to specify processes. You can use the `priocntl -i zoneid` command to move running processes into a different scheduling class in a non-global zone.

priocntl(2)

Added `P_ZONEID id` argument.

priv_str_to_set(3C)

Added "zone" string for the set of all privileges available within the caller's zone.

privileges(7)

Added `PRIV_PROC_ZONE`, which allows a process to trace or send signals to processes in other zones. See `zones(7)`.

proc(1)

Added `-z zone` option to `ptree` only. This option is only useful when the command is executed in the global zone.

proc(5)

Added capability to obtain information about processes running in zones.

processor_info(2)

If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.

prstat(8)

Added `-z zoneidlist` option. Also added `-Z` option.

If executed in a non-global zone in which the pools facility is enabled, the percentage of recent CPU time used by the process is displayed only for the processors in the processor set of the pool to which the zone is bound.

Output of the `-a`, `-t`, `-T`, `-J`, and `-Z` options displays a SWAP instead of a SIZE column. The swap reported is the total swap consumed by the zone's processes and `tmpfs` mounts. This value assists in monitoring the swap reserved by each zone, which can be used to choose a reasonable `zone.max-swap` setting.

ps(1)

Added `zonename` and `zoneid` to list of recognized `format` names used with the `-o` option.

Added `-z zonelist` to list only processes in the specified zones. Zones can be specified either by zone name or by zone ID. This option is only useful when the command is executed in the global zone.

Added `-Z` to print the name of the zone associated with the process. The name is printed under an additional column header, `ZONE`.

pset_bind(2)

Added `P_ZONEID` as `idtype`. Added `zone` to possible choices for `P_MYID` specification. Added `P_ZONEID` to valid `idtype` list in `EINVAL` error description.

pset_getloadavg(3C)

If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.

pset_info(2)

If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.

pset_list(2)

If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.

pset_setattr(2)

If the caller is in a non-global zone and the pools facility is enabled, but the processor is not in the processor set of the pool to which the zone is bound, an error is returned.

psrinfo(8)

If executed in a non-global zone, only information about the processors visible to the zone is displayed.

renice(1)

Added *zoneid* to list of valid arguments used with the *-i* option.

sar(1)

If executed in a non-global zone in which the pools facility is enabled, the *-b*, *-c*, *-g*, *-m*, *-p*, *-u*, *-w*, and *-y* options display values only for processors that are in the processor set of the pool to which the zone is bound.

sysconf(3C)

If the caller is in a non-global zone and the pools facility enabled, `sysconf(_SC_NPROCESSORS_CONF)` and `sysconf(_SC_NPROCESSORS_ONLN)` return the number of total and online processors in the processor set of the pool to which the zone is bound.

sysinfo(2)

Changed `PRIV_SYS_CONFIG` to `PRIV_SYS_ADMIN`.

traceroute(8)

Usage change. When specified from within a non-global zone, the *-F* option has no effect because the "don't fragment" bit is always set.

ucred_get(3C)

Added `ucred_getzoneid()` function, which returns the zone ID of the process or *-1* if the zone ID is not available.

umount(2)

`ENOENT` is returned if file pointed to by *file* is not an absolute path.

vmstat(8)

When run in a non-global zone in which the pools facility is enabled, statistics are reported only for the processors in the processor set of the pool to which the zone is bound. Applies to output from the *-p* option and the `page`, `faults`, and `cpu` report fields.

8

Migrating an Oracle Solaris Zone

A zone migration transfers an existing zone from one host system into a zone on another system. This chapter discusses the cold zone migration method used to migrate `solaris` zones.

This chapter covers the following topics:

- [Authorizing Non-Root Users to Perform Non-Global Zone Migrations](#)
- [Using Cold Migration to Migrate a solaris Zone](#)
- [Migrating a Zone From a System That Is Not Usable](#)
- [Archiving and Moving Non-Global Zones That Are Not Using Shared Storage](#)

Authorizing Non-Root Users to Perform Non-Global Zone Migrations

The global zone administrator can assign non-root users the rights to perform migrations. For the names and details of the rights profiles, see [Using Rights Profiles to Install and Manage Zones](#).

You can restrict the profiles and their associated authorizations to individual zones by setting the `auths` property on the `admin` resource in the zone configuration. Set the `admin` resource's `auths` property to `auths=migrate` to enable the user to perform all types of migrations for the zone, or `auths=migrate.cold` to enable only cold migration and warm migration. See [admin Resource Type in Oracle Solaris Zones Configuration Resources](#) for more information.



Note:

Setting the `auths` property in the zone configuration automatically enables the appropriate profiles for the user.

How to Authorize a User to Perform Cold Migration of an Individual Zone

This procedure assumes the user is already assigned rights and authorizations to create, modify, and delete zone configurations. See [Authorizing a User to Configure Zones on a System](#).

Perform this procedure to delegate a Zone Migration administrator to cold migrate a specific zone.

1. **Assume the root role.**

For more information, see [Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*](#).

2. Set the auths and user properties of the admin resource.

```
global$ zonecfg -z zonename
zonecfg:zonename> add admin
zonecfg:zonename:admin> set user=username
zonecfg:zonename:admin> set auths=migrate.cold
zonecfg:zonename:admin> end
zonecfg:zonename> commit
```

username is authorized for this zone only.

3. Verify the zone administrator's rights in the non-global zone and the global zone.

In this example, *jdoe* is the user and the zone name is *zone1*.

```
global$ zonecfg -z zone1 info admin
admin:
    user: jdoe
    auths: migrate.cold

global$ auths jdoe
solaris.admin.wusb.read,solaris.mail.mailq,solaris.network.autoconf.read,solaris.zone.migrate.cold/zonename

global$ profiles jdoe
jdoe:
Zone Cold Migration
Basic Solaris User
All
```

Example 8-1 Authorizing a User to Migrate All Zones on a System

This example sets authorization for user *jdoe* to perform cold migration of any zone on the source system.

```
global$ usermod -P +"Zone Migration" -A +solaris.zone.migrate jdoe
Verify the auths and profiles:

global$ auths jdoe
solaris.admin.wusb.read,solaris.mail.mailq,solaris.network.autoconf.read,solaris.zone.migrate
global$ profiles jdoe
jdoe:
Zone Migration
Basic Solaris User
All
```

Example 8-2 Authorizing a User to Configure Zones on a System

This example assigns the user *jdoe* the required profiles and authorizations needed to create, modify, and delete any zone configuration, then verifies the assignments. This assignment is necessary for the user to perform a cold migration.

```
global$ usermod -P +"Zone Configuration" -A +solaris.zone.config jdoe

global$ auths jdoe
solaris.admin.wusb.read,solaris.mail.mailq,solaris.network.autoconf.read,solaris.zone.config

global$ profiles jdoe
```

```
jdoe:
Zone Configuration
Basic Solaris User
All
```

Using Cold Migration to Migrate a `solaris` Zone

You can use cold migration to migrate a `solaris` zone that is configured to use shared storage. In a cold migration, a non-running zone in the `installed` state is detached, then moved and attached on another host where it is ready to be rebooted.

How to Cold Migrate a `solaris` Zone

- Review cold migration requirements in [Migration and Transformation of Non-Global Zones](#).
- Ensure that you have the necessary rights and privileges to cold migrate a zone. See [Authorizing Non-Root Users to Perform Non-Global Zone Migrations](#).

Perform this procedure cold migrate a `solaris` zone that uses shared storage.



Note:

If the zone uses local storage, you cannot migrate the zone using cold migration. Instead, use Unified Archives to migrate the zone. Go to [How to Use Unified Archive to Move a Non-Global Zone to a New System](#).

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. If the zone to be migrated is running, shut it down.

```
source-host$ pfbash zoneadm -z zonename shutdown
```

3. (Optional) Verify that the state is installed.

For example:

```
global$ zoneadm -z zone1 list -v
ID NAME          STATUS    PATH    BRAND    IP
- zone1         installed -       solaris  excl
```

4. (Optional) If you plan to use the `ssh://` URI to connect to the target host, test Secure Shell promptless authentication.

Execute a command such as `date` through `ssh` on the target host.

```
global1$ ssh global2 date
Mon Mar 9 13:22:40 PDT 2015
```

If you are prompted for a password, you have not configured your key pairs to enable login without interactive authentication.

See [How to Generate a Public/Private Key Pair for Use With Secure Shell in Managing Secure Shell Access in Oracle Solaris 11.4](#).

5. (Optional) Perform a dry run of the migration to verify that conditions are set appropriately.

```
source-host$ zoneadm -z zonename migrate -n ssh://user@target-host
```

For example:

```
global$ zoneadm -z zone1 migrate -n ssh://global2
zoneadm: zone 'zone1': Importing zone configuration.
zoneadm: zone 'zone1': Attaching zone.
zoneadm: zone 'zone1': Dry-run migration successful.
zoneadm: zone 'zone1': Cleaning up.
```

6. Perform the migration.

```
source-host$ zoneadm -z zonename migrate ssh://user@target-host
```

For example:

```
global$ zoneadm -z zone1 migrate ssh://global2
zoneadm: zone 'zone1': Importing zone configuration.
zoneadm: zone 'zone1': Attaching zone.
zoneadm: zone 'zone1': Migration successful.
```

7. (Optional) Boot the zone on the target host.

```
target-host$ zoneadm -z zonename boot
```

Example 8-3 Failing a Cold Migration Dry Run Due to Local Storage

This example verifies that a zone is not running. A cold migration dry run then fails because of local storage being used in the zone `zone2`.

```
root@global3 $ zoneadm list -cv
ID NAME           STATUS   PATH                               BRAND  IP
0  global          running  /                                   solaris shared
-  zone2           installed -                                   solaris excl
root@global3 $ zoneadm -z zone2 migrate -n ssh://global5
zoneadm: zone 'zone2': configuration check failed: The storage property
dev:/dev/zvol/dsk/rpool/VARSHARE/zones/zone2/disk0 is not a shared storage URI.
```

Example 8-4 Succeeding at Cold Migration After a Configuration Change

This example shows removal of a non-booting local storage device from the zone configuration for zone `zone2`, and then a successful migration. The zone configuration used comes from the migrating zone and is imported on the target host.

```
root@global3 $ zonecfg -z zone2 'remove device id=1;commit;exit'
root@global3 $ zoneadm -z zone2 migrate ssh://global5
zoneadm: zone 'zone2': Importing zone configuration.
zoneadm: zone 'zone2': Attaching zone.
zoneadm: zone 'zone2': Migration successful.
```

Migrating a Zone From a System That Is Not Usable

A system that hosts a non-global zone can become unusable. However, if the zone is on shared storage that is still usable, you might still be able to migrate the zone to a new host successfully by moving the `zonepath` for the zone to the new host.

In some cases, such as a SAN, the `zonepath` data might not actually move. The SAN might simply be reconfigured so the `zonepath` is visible on the new host.

Since the unusable zone was not properly detached, you must first create the zone on the new host by using the `zonecfg` command. After the zone is created on the new host, you can attach the zone to the host.

Archiving and Moving Non-Global Zones That Are Not Using Shared Storage

If a `solaris` zone is not using shared storage for its zone data, it cannot be migrated by using the `zoneadm migrate` command. Instead, archive the zone and then deploy the zone on a new host.

How to Use Unified Archive to Move a Non-Global Zone to a New System

Perform this procedure to transform a `solaris` zone that is not on shared storage.

Note:

If the zone is configured on shared storage, go to [Moving Existing Zones To or From Shared Storage Zone Configurations](#).

1. **On the global zone, become an administrator.**

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. **Create a recovery archive of the source system's non-global zone that you want to migrate.**

```
source$ pfbash archiveadm create -r -z zonename  
zone-archive-name.uar
```

3. **Uninstall the non-global zone on the source system, or set the `autoboot` property for the zone to `false`.**

- To uninstall the zone, perform the following command:

```
source$ zoneadm -z zonename uninstall
```

- To disable the `autoboot` property, perform the following command:

```
source$ zonecfg -z zonename set autoboot=false
```

4. **Configure the non-global zone using the archive on the target host.**

```
target$ pfbash zonecfg -z zonename create -a /path/zone-archive-name.uar
```

5. **Install the zone on the target host from the archive.**

```
target$ zoneadm -z zonename install -a zone-archive-name.uar
```

6. **Boot the installed zone on the target host.**

```
target$ zoneadm -z zonename boot
```

See Also

For additional information about creating and deploying Unified Archives, refer to [Chapter 2, Working With Unified Archives in *Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4*](#).

How to Move a Zone That Is Not on Shared Storage

Perform this procedure to move a zone to a new location on the same system by changing the `zonopath`. The zone must first be halted. The normal `zonopath` criteria are described in [Zone Global Properties in *Oracle Solaris Zones Configuration Resources*](#).

Note:

Observe the following restrictions on moving zones that are not on shared storage:

- You cannot move a zone that is present in other BEs. You can either delete those BEs first, or create a new zone at the new path by cloning the zone.
- For zone configurations that use a `rootzpool` resource, the `zonopath` property is fixed and immutable, based on the default `zonopath` template value. Manually setting the `zonopath` to a value other than the derived default value is prohibited in this zone configuration.

To move a zone that is on shared storage, instead go to [Moving Existing Zones To or From Shared Storage Zone Configurations](#).

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Halt the zone to move.

```
global$ zoneadm -z zonename halt
```

3. Move the zone to a new zonopath.

```
global$ zoneadm -z zonename move /new-zonopath/zonename
```

4. Verify the path.

```
global$ zoneadm list -cv
ID  NAME      STATUS    PATH                                BRAND  IP
0   global    running  /                                  solaris shared
-   zonename installed /new-zonopath/zonename        solaris excl
```

9

Live Zone Reconfiguration of Oracle Solaris Zones

Use Live Zone Reconfiguration to reconfigure or report on the live configuration of Oracle Solaris zones while the zones are running.

To view a table that shows Live Zone Reconfiguration support for resources and properties in `solaris` zones and kernel zones, see the table in [Live Zone Reconfiguration Support or Restriction in Oracle Solaris Zones Configuration Resources](#).

Live Zone Reconfiguration of Oracle Solaris Zones

Use Live Zone Reconfiguration to reconfigure or report on the live configuration of Oracle Solaris zones while the zones are running.

This section covers the following information:

- [zonecfg Utility Edit Modes](#)
- [Temporary Changes to the Live Zone Configuration](#)
- [Persistent Changes to the Live Zone Configuration](#)
- [Live Zone Reconfiguration Dry Run](#)
- [Reloading a Live Zone Configuration](#)

`zonecfg` Utility Edit Modes

Use the `zonecfg` utility edit modes to make configuration changes to the zone. You can change either the persistent stored configuration or the running live configuration. The `zonecfg` utility supports the following edit modes for use with Live Reconfiguration:

Default mode

Create, modify, and list the persistent zone configuration stored on the stable storage. Parameters you changed in the default mode do not affect a running zone at the time you make the changes. The default mode is the primary way to maintain the zone configuration. This mode is backward compatible.

To have the changes made in default mode take effect in the running zone, you must issue one of the following `zoneadm` commands:

- Use the `zoneadm apply` command to load the updated persistent zone configuration so it is applied to the running zone.
- Use the `zoneadm reboot` command to reboot the zone and read the updated persistent zone configuration.

Live mode

Retrieve, inspect and edit the running live zone configuration. The live mode is available for a running zone only. Parameters you change in live mode take effect immediately after you use the `commit` subcommand to enable them in the live zone configuration.

Changes made in live mode are temporary. The changes remain active until the next zone reboot. For more information, see [Temporary Changes to the Live Zone Configuration](#).

To make live zone configuration changes permanent, you apply the changes to the persistent zone configuration with the `zoneadm apply` command. For more information, see [Persistent Changes to the Live Zone Configuration](#).

To enable live mode, use the `-r` option with the `zonecfg` command, which retrieves the live zone configuration instead of the persistent zone configuration.

```
global$ pfexec zonecfg -z zonename -r
```

You can work with the `zonecfg -r` command just as you do in default mode. The full set of `zonecfg` subcommands and both the interactive and the batch mode are supported.

Not all resources can be reconfigured in the live configuration. For a list of supported resource types and properties, see the table in [Live Zone Reconfiguration Support or Restriction in Oracle Solaris Zones Configuration Resources](#).

Temporary Changes to the Live Zone Configuration

You might want to make only temporary changes in the configuration of a running zone. For example, you might want to remove a resource from a zone temporarily while maintenance is performed on a device, or allocate a resource temporarily for a special purpose but not have it be present for the entire run of the zone. Such removals or additions of resources should not be done in the persistent zone configuration because they would cause a failure when the zone reboots and the resource is no longer available.

Parameters changed temporarily in live mode take effect immediately after you issue the `zonecfg commit` command. These changes are valid until the next zone reboot.

Persistent Changes to the Live Zone Configuration

You use the `zoneadm apply` command to apply changes from the persistent zone configuration to the live zone configuration. You do not have to reboot for the changes to affect the running zone. For an example, see [How to Make Persistent Configuration Changes to a Live Zone](#).

Live Zone Reconfiguration Dry Run

You can test run the effects of changes to the live zone configuration before putting those changes into effect by using the following options to the `zonecfg commit` and `zoneadm apply` commands:

-n

Dry run mode. The command shows the effects of the changes to the configuration, but applies no changes to the running zone. Use the dry run mode to preview the

actions that would be performed if you issued the `zonecfg commit` or `zoneadm apply` command to impact the live zone configuration.

-q

Quiet mode. This mode suppresses all system messages and returns a status code only.

Reloading a Live Zone Configuration

If the configuration of a running zone changes externally while you are modifying the configuration in either default mode or live mode, the `zonecfg commit` command will return an error. Some scenarios where this might occur include another administrator modifying the configuration, modifying resource controls, or changing network parameters of the zone using network administration commands.

If the configuration of a running zone changes during live reconfiguration, use the `zonecfg reload` subcommand to load the external configuration changes:

- If you issue the `zonecfg reload` command in default mode, the command discards any uncommitted changes you have made and reloads the configuration from persistent storage.
- If you issue the `zonecfg reload` command in live mode (the `-r` option), the command discards any uncommitted changes and retrieves an up-to-date live configuration of the running zone.

After the configuration is reloaded, you can repeat the configuration changes and commit.

See [How to Recover From a Failure While Committing Live Zone Configuration Changes](#) for an instructions to reload a zone configuration.

Performing a Live Zone Reconfiguration

This section provides the following procedures to perform common live zone reconfiguration tasks:

- [How to Inspect the Live Configuration of a Running Zone](#)
- [How to Preview the Effect of a Live Zone Configuration](#)
- [How to Make Persistent Configuration Changes to a Live Zone](#)
- [How to Make Temporary Changes to the Running Zone](#)
- [How to Recover From a Failure While Committing Live Zone Configuration Changes](#)

How to Inspect the Live Configuration of a Running Zone

Perform this procedure to view and export the configuration of a running zone.

- 1. On the global zone, become an administrator.**

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

- 2. Display information about the live zone configuration.**

```
global$ pfbash zonecfg -z zonename -r info
```

- 3. (Optional) Export the live configuration.**

```
global$ zonecfg -z zonename -r export -f exported.cfg
```

How to Preview the Effect of a Live Zone Configuration

Review the following:

- [Live Zone Reconfiguration Dry Run](#)
- [Appendix A, Resource Types and Global Properties That Support Live Zone Reconfiguration in Oracle Solaris Zones Configuration Resources](#)

Perform this procedure to review the live zone configuration changes that would be made, before you make a final commitment of those changes.

1. On the global zone, become an administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Start the zonecfg utility in live mode and configure the desired zone changes.

```
global$ pfbash zonecfg -z zonename -r
zonecfg:zonename> Make zone configuration changes
```

3. View the actions that would be performed by the reconfiguration.

The -n option prevents actual commitment of the zone changes.

```
zonecfg:zonename> commit -n
```

Next Steps

To make the previewed changes to the live zone configuration, issue the `zonecfg commit` command without using the -n option.

How to Make Persistent Configuration Changes to a Live Zone

Review the following documentation:

- [zonecfg Utility Edit Modes](#)
- [Persistent Changes to the Live Zone Configuration](#)
- [Appendix A, Resource Types and Global Properties That Support Live Zone Reconfiguration in Oracle Solaris Zones Configuration Resources](#)

Perform this procedure to make live zone configuration changes that persist across reboots of the zone.

1. On the global zone, become an administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Make changes to the zone in default mode.

```
global$ pfbash zonecfg -z zonename "zonecfg-commands"
```

Specify one or more `zonecfg` subcommands.

```
zonename> "zonecfg-commands"
```

For the format, see the [zonecfg\(8\)](#) man page.

3. Apply the changes to the live configuration.

```
global$ pfbash zoneadm -z zonename apply
```

Example 9-1 Reducing the Number of Virtual CPUs in the Live Zone Configuration

This example shows a running kernel zone `kz1` that has 16 virtual CPUs configured. The persistent configuration is changed to set the number of VCPUs to 8 and applied to the live configuration. The output shows what happens if the kernel zone cannot stop using some of the CPUs. This might occur if the kernel zone contains a `solaris` zone that is configured with the `dedicated-cpu` resource type, for instance.

The live zone reconfiguration tries to satisfy the request by skipping those CPUs that cannot be removed from the kernel zone (the guest) while trying to remove others. When the specified number of CPUs cannot be removed, the operation succeeds partially and the output shows the new number of virtual CPUs.

```
global$ pfbash zonecfg -z kz1 -r info virtual-cpu
virtual-cpu:
    ncpus: 16
global$ zonecfg -z kz1 'select virtual-cpu;set ncpus=8;end'
global$ zoneadm -z kz1 apply
zone 'kz1': Checking: Modifying virtual-cpu ncpus=8
zone 'kz1': Applying the changes
zone 'kz1': error: dr-cpu failed for cpu id=15: Operation was blocked
zone 'kz1': error:          status: CPU is configured for use by the guest
zone 'kz1': error: dr-cpu failed for cpu id=14: Operation was blocked
zone 'kz1': error:          status: CPU is configured for use by the guest
...
operation continues to try to remove 8 virtual CPUs
...
zone 'kz1': warning: operation succeeded partially for virtual cpus (requested: 8,
final: 12)

global$ zonecfg -z kz1 -r info virtual-cpu
virtual-cpu:
    ncpus: 12
```

How to Make Temporary Changes to the Running Zone

Review the following documentation:

- [zonecfg Utility Edit Modes](#)
- [Temporary Changes to the Live Zone Configuration](#)
- [Appendix A, Resource Types and Global Properties That Support Live Zone Reconfiguration in *Oracle Solaris Zones Configuration Resources*](#)

Perform this procedure to temporarily change the live configuration of a running zone and then restore the persistent configuration to undo the change.

1. On the global zone, become an administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Change the zone configuration.

The sample command adds a disk in live mode and shows the command output.

 **Note:**

The `zonecfg commit` command is not required. The `zonecfg` utility commits the changes when the command exits.

```
global$ pfbash zonecfg -z kzone -r 'add device;set storage=dev:/dev/dsk/
cNtXd;end'
zone 'kzone': Checking: Adding device storage=dev:/dev/dsk/cNtXd
zone 'kzone': Applying the changes
```

3. (Optional) When you no longer need the configuration change, return the zone to the persistent zone configuration.

The sample command removes the temporary configuration change.

```
global$ zoneadm -z kzone apply
zone 'kzone': Checking: Removing device storage=dev:/dev/dsk/cNtXd
zone 'kzone': Applying the changes
```

Alternatively, you can reboot the zone to discard the live zone configuration changes and return to the persistent zone configuration.

Troubleshooting

If the `commit` operation reports an error, see [How to Recover From a Failure While Committing Live Zone Configuration Changes](#).

How to Recover From a Failure While Committing Live Zone Configuration Changes

The configuration of a running zone can change externally while a live zone configuration is being edited. When this conflict occurs, the `zonecfg commit` command returns an error.

Perform this procedure to correct the error by reloading the zone configuration to show the updated version and then making your edits again.

1. On the global zone, become an administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Issue the `reload` subcommand, make the correct configuration changes, and commit the change.

This step assumes that you are still in the `zonecfg` session that failed to commit your temporary configuration changes.

```
zonecfg:zonename> reload
zonecfg:zonename>
temporary-configuration-changes
zonecfg:zonename> commit
```

Example 9-2 Recovering From Failed Temporary Zone Configuration Changes

The following example shows temporary configuration changes, an error message that the changes failed, recovery steps, and output confirming that the changes now succeeded.

```

Example configuration change in live mode
global$ pfbash zonecfg -z kzone1 -r
zonecfg:kzone1> add anet;set lower-link=net1;set maxbw=2G;end
zonecfg:kzone1> commit
zone 'kzone1': error: the live configuration has changed externally.
Trying to commit changes to externally changed live configuration
Please use reload to start again. Your local changes will be lost.

Reload the configuration
zonecfg:kzone1> reload
Are you sure you want to reload (y/[n])? y
Repeat the configuration changes you previously attempted
zonecfg:kzone1> add anet;set lower-link=net1;set maxbw=2G;end
zonecfg:kzone1> commit
Command output shows the configuration changes now succeed
zone 'kzone1': Checking: Adding anet id=2
zone 'kzone1': Applying the changes
...

```

Live Zone Reconfiguration of Datasets

Use Live Zone Reconfiguration of datasets in the following cases:

- In the global zone, to remove a zpool dataset resource from a running non-global zone.
- In the non-global zone, to import a new virtual zpool. The zpool is made available to the zone by the global zone administrator, who adds the dataset resource through Live Zone Reconfiguration.

This section provides the following information:

- [How to Add a ZFS Dataset to a Running Zone](#)
- [How to Remove a dataset Resource From a Running Zone](#)
- [How to Export a Dataset From a solaris Zone](#)

How to Add a ZFS Dataset to a Running Zone

Perform this procedure to add ZFS datasets to a running zone.



Note:

Issue all commands in this procedure from the global zone.

1. On the global zone, become an administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Display the current dataset information.

```

global$ pfbash zonecfg -z zonename info dataset
dataset:
    name: rpool/export/dataset1
    alias: dataset1
dataset:

```

```
name: rpool/export/dataset2
alias: dataset2
```

3. Inspect the current live configuration and datasets for the zone.

```
global$ zonecfg -z zonename -r
zonecfg:zonename> info dataset
dataset 0:
    name: rpool/export/dataset2
    alias: dataset2
dataset 1:
    name: rpool/export/dataset1
    alias: dataset1
zonecfg:zonename> exit
```

4. Create a new dataset.

```
global$ zfs create rpool/export/new-dataset
global$ zfs unmount rpool/export/new-dataset
```

5. Add the new dataset resource to the running solaris zone.

Verify the addition, commit the change, and exit the zonecfg utility.

```
global$ zonecfg -z zonename -r
                                Add the dataset
zonecfg:zonename> add dataset
zonecfg:zonename:dataset> set name=rpool/export/new-dataset
zonecfg:zonename:dataset> end
                                Verify the added dataset
zonecfg:zonename> info dataset
dataset 0:
    name: rpool/export/dataset2
    alias: dataset2
dataset 1:
    name: rpool/export/dataset1
    alias: dataset1
dataset 2:
    name: rpool/export/new-dataset
    alias: new-dataset
                                Commit the change
zonecfg:zonename> commit
zone 'zonename': Checking: Adding dataset name=rpool/export/new-dataset
zone 'zonename': Checking: Adding dataset alias=new-dataset
zone 'zonename': Applying the changes
Exit the command
zonecfg:zonename> exit
```

Note:

The dataset is not yet added to the persistent stored zone configuration, only to the live zone configuration.

6. Change the persistent zone configuration to match the live zone configuration.

```
global$ zoneadm -z zonename apply
```

7. Verify that the new dataset is added to the persistent zone configuration.

```
global$ zonecfg -z zonename info dataset
dataset:
```

```

        name: rpool/export/new-dataset
        alias: new-dataset
dataset:
    name: rpool/export/dataset1
    alias: dataset1
dataset:
    name: rpool/export/dataset2
    alias: dataset2

```

How to Remove a `dataset` Resource From a Running Zone

Perform this procedure to remove ZFS datasets from a running zone.

Before you can remove the `dataset` `dataset`, you must first use the `zpool export` command to export the virtual zpool while in the non-global zone.

- 1. On the global zone, become an administrator.**

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

- 2. View the dataset information of the live zone configuration.**

```

zonename$ pfbash zonecfg -z zonename -r info dataset
...
dataset:
    name: rpool/export/dataset1
    alias: dataset1
dataset:
    name: rpool/export/dataset2
    alias: dataset2
...

```

- 3. Remove the dataset from the running zone, verify the change, and commit the change to the live zone configuration.**

```

zonename$ zonecfg -z zonename -r
zonecfg:zonename> remove dataset name=rpool/export/dataset1
zonecfg:zonename> info dataset
dataset 0:
    name: rpool/export/dataset2
    alias: dataset2
...
zonecfg:zonename> commit
zone 'zonename': Checking: Removing dataset name=rpool/export/dataset1
zone 'zonename': Applying the changes
zonecfg:zonename> exit

```

- 4. Display the dataset information of the live zone configuration.**

Verify that the dataset is removed from the live zone configuration.

```

zonename$ zonecfg -z zonename -r info dataset
dataset:
    name: rpool/export/dataset2
    alias: dataset2

```

- 5. (Optional) Change the persistent zone configuration to match the live zone configuration.**


```
zonename$ zoneadm -z zonename apply
```

How to Export a Dataset From a `solaris` Zone

This procedure exports a dataset for migration or move. If the dataset is in use, the export might not succeed. For more information, see [Exporting a ZFS Storage Pool in *Managing ZFS File Systems in Oracle Solaris 11.4*](#).

1. On the global zone, become an administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. (Optional) Display the zone datasets that could be exported.

- Display the zone datasets from the non-global zone.

```
zonename$ pfbash zonecfg -z zonename info dataset
dataset:
    name: rpool/export/dataset1
    alias: dataset1
dataset:
    name: rpool/export/dataset2
    alias: dataset2
```

- Log in to the zone, then display dataset information.

```
$ zlogin zonename
zonename$ zpool list
NAME      SIZE  ALLOC   FREE   CAP  DEDUP  HEALTH  ALTROOT
dataset1  68G   21.5G  46.5G  31%  1.00x  ONLINE  -
dataset2  68G   21.5G  46.5G  31%  1.00x  ONLINE  -
rpool     68G   21.5G  46.5G  31%  1.00x  ONLINE  -

zonename
zfs list -t all|grep ds
dataset1                                31K  45.4G   31K  /dataset1
dataset2                                31K  45.4G   31K  /dataset2
zonename
mount -v|grep ds
dataset1 on /dataset1 type zfs read/write/setuid/nodevices/rstchown/
nonbmand/exec/xattr/atime/
zone=zonename/sharezone=1/dev=3350020 on Thu Jun 11 08:40:30 2015
dataset2 on /dataset2 type zfs read/write/setuid/nodevices/rstchown/
nonbmand/exec/xattr/atime/
zone=zonename/sharezone=1/dev=3350021 on Thu Jun 11 08:40:30 2015
```

3. From the non-global zone, export the dataset.

```
zonename$ zpool export dataset1
```

4. List the running zpools.

The exported dataset should not be listed.

```
zonename$ zpool list
NAME      SIZE  ALLOC   FREE   CAP  DEDUP  HEALTH  ALTROOT
dataset2  68G   21.5G  46.5G  31%  1.00x  ONLINE  -
rpool     68G   21.5G  46.5G  31%  1.00x  ONLINE  -
```

10

Transforming Systems to Oracle Solaris Zones

This chapter describes zone conversions and shows how to convert physical systems to zones and convert between different types of zones. Zone conversion means one of the following transformations:

- Transforming global zones into non-global zones, also known as physical to virtual (P2V) conversion
- Transforming non-global zones into global zones
- Transforming `solaris` zones into kernel zones

Transformation involves data transfer using Oracle Solaris Unified Archives. Along with zone migration, transforming zones using archives can be part of a strategy to move existing systems for server consolidation, workload rebalancing, and disaster recovery.

For background information, review the following:

- [About Zone Transformations](#)
- [About Zone Migration in *Introduction to Oracle Solaris Zones*](#)

Transforming a Global Zone Into a `solaris` Non-Global Zone

This section provides procedures that use Unified Archives to transform an existing Oracle Solaris 11.4 system into a `solaris` brand zone on an Oracle Solaris 11.4 system.

- [How to Check the Source System With `zonep2vchk`](#)
- [How to Create an Archive of the System Image](#)
- [How to Configure the Transformed Zone on the Target System](#)
- [How to Install the Transformed Zone From the Archive on the Target System](#)

How to Check the Source System With `zonep2vchk`

Review planning information and guidelines in [About Zone Transformations](#) and [Using the `zonep2vchk` Tool to Prepare for Global to Non-Global Transformations](#).

Perform this procedure on the source system.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Perform a basic analysis that checks for Oracle Solaris features in use that might be impacted by a transformation.

The `-b` option specifies a basic analysis.

```
source$ zonep2vchk -b
```

3. Perform a static analysis of application files.

The `-s` option specifies a static analysis. The option takes a comma-separated list of paths. The following example command inspects two application paths for ELF binaries for system and library calls that might affect operation inside a zone.

```
source$ zonep2vchk -s /opt/myapp/bin,/opt/myapp/lib
```

4. Perform runtime checks that look for processes that could not be executed successfully inside a zone.

The `-r` option specifies a runtime check. The following example command performs a runtime check for two hours.

```
source$ zonep2vchk -r 2h
```

5. Generate a template zonecfg configuration.

The template name you specify must end with the `.config` extension.

```
source$ zonep2vchk -c > /path/filename.config
```

This configuration will contain resource limits and network configuration based on the physical resources and networking configuration of the source host.

Next Steps

Go to [How to Create an Archive of the System Image](#).

How to Create an Archive of the System Image

Perform this procedure to archive the file systems of the global zone that you will transform to a `solaris` zone.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Verify that no non-global zones are installed on the source system, including kernel zones.

A recovery archive that contains non-global zones would fail to install in a `solaris` zone.

```
global$ zoneadm list -cv
  ID NAME           STATUS   PATH                               BRAND  IP
   0 global         running  /                                   solaris
shared
```

If there are non-global zones or kernel zones on the system, delete them by following procedures in [Uninstalling and Deleting a Non-Global Zone](#).

3. Create a recovery archive of the global zone on network storage.

Specify the fully-qualified path of the recovery archive to create and end the file name with the `.uar` extension.

```
source$ pfbash archiveadm create -r -z global recovery-archive.uar
```

Next Steps

The Unified Archive generated with the `archiveadm` command contains a zone configuration that serves as a starting point for the converted zone. Additional

configuration might be needed to configure the zone in a way that is optimal for the target system.

For more information, see [Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4](#) and the `archiveadm(8)` man page.

Troubleshooting

Tip:

In some deployments, data for the operating system and applications are segregated into separate ZFS pools. In this case, a backup and recovery system other than Unified Archives is used for protecting application data. You might want to create root-only archives. A root-only archive includes only the root pool, typically `rpool`. The action is the equivalent of using `--exclude-dataset` for each ZFS pool that is not the root pool.

```
source$ pfbash archiveadm create -z global --recovery --root-only /archives/  
server1.uar
```

Next Steps

Go to [How to Configure the Transformed Zone on the Target System](#).

How to Configure the Transformed Zone on the Target System

The template `zonecfg` script generated by the `zonep2vchk` tool defines aspects of the source system's configuration that must be supported by the target system's zone configuration. To fully configure the zone, perform this procedure to provide additional information specific to the target system.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Configure the zone from the archive.

```
target$ zonecfg -z zonename  
Use 'create' to begin configuring a new zone.  
zonecfg:zonename> create -a /path/config-file.uar  
zonecfg:zonename> info
```

3. Add the storage resources for the zone.

This example command shows how to configure shared storage for the `rootzpool`, which is recommended when configuring zones deployed using Unified Archives. For more information, see [Oracle Solaris Zones on Shared Storage](#).

```
zonecfg:zonename> add rootzpool  
zonecfg:zonename:rootzpool> add storage iscsi://zfssa/  
luname.naa.600144F0DBF8AF19000052E820D60003  
zonecfg:zonename:rootzpool> end
```

Do not yet exit the `zonecfg` utility.

4. From the global zone of the target system, view the current link configuration to verify that it is correct.

```
target$ dladm show-link
target$ dladm show-phys
target$ ipadm show-addr
```

 **Note:**

By default, the `zonecfg` template defines a network configuration with an `anet` resource for every physical network interface that was configured on the source system. The target system automatically creates a VNIC for each `anet` resource when the zone boots. The use of VNICs make it possible for multiple zones to share the same physical network interface. The lower-link name of an `anet` resource is initially set to *change-me* by the `zonecfg` command. You must manually set this field to the name of one of the data links on the target system. Any link that is valid for the lower-link of a VNIC can be specified.

5. Commit and exit the zone configuration.

```
zonecfg:zonename> commit
zonecfg:zonename> exit
```

Next Steps

Go to [How to Install the Transformed Zone From the Archive on the Target System](#).

How to Install the Transformed Zone From the Archive on the Target System

Perform this procedure to install the transformed zone. This procedure does not alter the original system configuration during the installation.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Install the zone by using the Unified Archive file that you created on the source system.

```
target$ zoneadm -z zonename install -a config-file.uar
```

11

Configuring and Administering Immutable Zones

This chapter describes how to configure immutable zones by specifying the zone's security policy and how to maintain immutable zones by configuring administrative access.



Note:

All zones can be made immutable: the physical global zone, non-global zones, and virtual global zones (called kernel zones).

This chapter covers the following topics:

- [About Immutable Zones](#)
- [Configuring Immutable Zones](#)
- [Administering Immutable Non-Global Zones](#)
- [Immutable Global Zones](#)

About Immutable Zones

An immutable zone is a zone with a read-only `root` file system. The zone can be non-global or global. The read-only root preserves the zone's configuration. Also, additional restrictions to the runtime environment extend the zone's secure runtime boundary. Maintenance operations are possible, but you as administrator must take deliberate steps to access the zone for maintenance. The mandatory write access control (MWAC) security policy blocks modifications to system binaries or system configurations.

MWAC is used to enforce file system write privilege through an SMF property, `file-mac-profile`. You can specify the MWAC security policy by modifying the `file-mac-profile` value with the `zonecfg` command. The policy is enforced in the kernel. Because the global zone is not subject to the MWAC policy of a non-global zone, the global zone can write to a non-global zone's file system for installation, image updates, and maintenance.

The MWAC policy is downloaded when the zone enters the `ready` state. The policy is enabled at zone boot. To perform post-install assembly and configuration, a temporary writable root-file system boot sequence is used. Modifications to the zone's MWAC configuration only take effect when you reboot the zone.

Configuring Immutable Zones

Mutable and immutable zones are differentiated by their MWAC security policy, which you specify with the `file-mac-profile` property of the `zonecfg` command.

Setting the MWAC Security Policy

By default, the `file-mac-profile` property is not set and the zone has a writable root dataset.

Several values for `file-mac-profile` restrict access to all or part of the runtime environment from inside the zone. All of the profiles except `none` will cause the `/var/pkg` directory and its contents to be read-only from inside the zone. The `none` MWAC security policy is equivalent to an unset MWAC security policy.

The following MWAC values restrict access to all or part of the runtime environment from inside the zone:

dynamic-zones

Is valid for global zones, including the global zone of a kernel zone. Permits the creation and the destroying of kernel zones and non-global zones.

Is equivalent to `fixed-configuration`, but adds the ability to create and destroy kernel zones and non-global zones.

Is similar to `flexible-configuration`, but `dynamic-zones` cannot write to files in the `/etc` directory.

fixed-configuration

Permits updates to `/var/*` directories, with the exception of directories that contain system configuration components.

- IPS packages, including new packages, cannot be installed.
- Persistently enabled SMF services are fixed.
- SMF manifests cannot be added from the default locations.
- Logging and auditing configuration files can be local. `syslog` and audit configuration are fixed.

flexible-configuration

Permits modification of files in `/etc/*` directories, changes to root's home directory, and updates to `/var/*` directories. This configuration provides the closest functionality to the Oracle Solaris 10 `native` sparse root zone documented in the Oracle Solaris 10 guide, *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones*.

- IPS packages, including new packages, cannot be installed.
- Persistently enabled SMF services are fixed.
- SMF manifests cannot be added from the default locations.
- Logging and auditing configuration files can be local. `syslog` and audit configuration can be changed.

strict

Read-only file system, no exceptions.

- IPS packages cannot be installed.
- Persistently enabled SMF services are fixed.
- SMF manifests cannot be added from the default locations.

- Logging and auditing configuration files are fixed. Data can only be logged remotely.
- Running an NFS server inside an immutable zone with this profile is not supported. You must use the `fixed-configuration` profile to run an NFS server.

Example 11-1 Setting the MWAC Security Policy for the Global Zone

In this example, you are assigned the Zone Security rights profile and create an immutable global zone. In this zone, the zone administrator can create and destroy kernel and non-global zones. Otherwise, the zone is immutable.

```
global$ zonecfg -z global set file-mac-profile=dynamic-zones
```

After the MWAC security policy is set and you reboot the immutable zone, the zone boots transient read-write until it reaches the `self-assembly-complete` milestone and then reboots in read-only mode.

Zone Resource Exceptions to MWAC Security Policy

Datasets that you add to a zone through the `zonecfg add dataset` command are not subject to MWAC policy. Zones have full control over added datasets. The platform datasets are visible, but their data and their properties are read-only unless the zone is booted read/write.

File systems that you add to a zone through the `zonecfg add fs` command are not subject to MWAC policy. To maintain the policy, mount the file systems read-only.

SMF Services in Immutable Zones

In an immutable zone, SMF services cannot be modified persistently by default, even when you have the authorizations required by that service. To enable or disable a service or change the value of a property, you must use the `-t` (temporary) option; the change is reverted when the zone is rebooted. If you do not use the `-t` option, you receive a "Permission denied" error message.

To make service changes that persist across reboots of the immutable zone, you must be operating in the Trusted Path Domain (TPD), as described in [Administering an Immutable Zone by Using the Trusted Path Domain](#). When you are operating in the TPD, you can make persistent changes to any service for which you have the required authorizations.

Some services can be configured so that the processes started by their methods run in the TPD. Services that have a `trusted_path` attribute set to `true` run in the TPD.

Caution:

The setting `trusted_path` to `true` makes the system less immutable. You are loosening the security that you wanted when you created the immutable zone.

In particular, do not set `trusted_path` to `true` for services that can be accessed from outside the zone. Specifically, interactive login services such as SSH should not be on the Trusted Path.

The `rad:remote` service described in [How to Enable Remote Administrative Access to an Immutable Zone by Using RAD](#) is an exception to this caution because the RAD API checks

Oracle Solaris authorizations and restricts what can be done. The `rad:remote` service provides no interactive shell and has strong authentication and authorization (`tpd=yes` in the `user_attr` entry for the user) over a secure transport to get to the host's TPD.

The `puppet:agent` service is another exception to the above caution. Updating the system by using Puppet should be trustworthy, rather than updating through an interactive login. Note that the `puppet:master` instance should not need to be on the Trusted Path.

The following example shows how to set the value of `trusted_path` so that the service runs in the TPD.

Example 11-2 Adding the Puppet Service to the Trusted Path

Check whether `trusted_path` is set for the service that you want to run in the TPD.

```
# svcprop -p method_context/trusted_path puppet:agent
svcprop: Couldn't find property group `method_context/trusted_path' for instance
`svc:/application/puppet:agent'.
```

The service shown in the preceding command is not running in the TPD because the `trusted_path` attribute is not set. The same would be true if the value of `trusted_path` was `false`.

To configure the `puppet:agent` service to run in the TPD, set `trusted_path` to `true`. Because this attribute does not exist for this service, you must specify the attribute type as well as the value. Refresh and restart the service after you set the attribute. The following commands include a verification step.

```
# svccfg -s puppet:agent
svc:/application/puppet:agent> setprop method_context/trusted_path = boolean:
true
svc:/application/puppet:agent> listprop method_context/trusted_path
method_context/trusted_path boolean      true
svc:/application/puppet:agent> refresh
svc:/application/puppet:agent> restart
svc:/application/puppet:agent> exit
# svcprop -p method_context/trusted_path puppet:agent
true
```

The following output shows information about the `puppet` process started by the `puppet:agent` service. See the [getpflags\(2\)](#) man page for descriptions of the TPD flags.

```
$ svcs -p puppet:agent
STATE          STIME      FMRI
online         15:05:50  svc:/application/puppet:agent
               15:05:50      7008 puppet

$ ppriv 7008
7008:      /usr/ruby/2.1/bin/ruby /usr/sbin/puppet agent --logdest /var/log/puppet
flags = PRIV_PROC_TPD|PRIV_TPD_UNSAFE|PRIV_TPD_KILLABLE
E: all
I: basic
P: all
L: all
```

Administering Immutable Non-Global Zones

This section provides information about how to determine whether a non-global zone is immutable. If the zone is immutable, administration requires some setup.

Note:

If you do not configure administrative access, you can administer the on-disk non-global zone from the global zone only. Within a running non-global zone, you can only administer the runtime state of an immutable zone. Therefore, modifying MWAC policy in a running zone is temporary. For more information, see [SMF Services in Immutable Zones](#).

Determining Whether a Non-Global Zone Is Immutable

The parsable output of the `zoneadm list -p` command from the global zone displays an R/W column and a file-mac-profile column. In the following output, the `fixed-configuration` `running` zones, `testzone2` and `testzone3`, are read-only, while the `testzone1` `running` zone is a read-write `fixed-configuration` zone.

```
global$ zoneadm list -p
0:global:running:/:UUID:solaris:shared:-:none
5:testzone2:running:/export/zones/testzone2:UUID \
:solaris:shared:R:fixed-configuration
12:testzone3:running:/export/zones/testzone3:UUID \
:solaris:shared:R:fixed-configuration
13:testzone1:running:/export/zones/testzone1:UUID \
:solaris:excl:W:fixed-configuration
-:testzone:installed:/export/zones/testzone:UUID \
:solaris:excl:-:fixed-configuration
```

Methods for Administering Non-Global Immutable Zones

Administrators must explicitly configure access to immutable zones for administrative operations. More secure methods require authorization and depend on the MWAC policy in effect and on your mode of access. Secure methods use the **trusted path**. A simple, insecure method is to briefly make the zone mutable, make your changes, then reboot the zone as immutable.

- **Use the Trusted Path Domain** – You must have access to a console and configure the console and users to access the Trusted Path Domain (TPD).

See [Administering an Immutable Zone by Using the Trusted Path Domain](#).

Note:

Except for the `flexible-configuration` policy, the other three MWAC policies enforce the `safe` mode, where you can access and modify immutable files only.

- **Make the entire zone temporarily writable** – You must have access to a terminal window and be authorized to run the `zoneadm` or `zlogin` command.

The `zoneadm` method is useful for small, fast fixes. You boot the immutable zone as temporarily writable, make your changes, and reboot.

The `zlogin` method is useful for editing protected files and updating packages. During the `zlogin` session, the zone is writable. This method cannot be used with console login.

See [Administering an Immutable Zone by Making It Writable](#).

Administering an Immutable Zone by Using the Trusted Path Domain

Oracle Solaris provides four ways to enter an immutable zone to administer it. Two methods make the entire zone temporarily writable, as described in [Methods for Administering Non-Global Immutable Zones](#). A safer mode uses the *trusted path*, in which only processes marked as part of the trusted path can be modified while the files and other zone processes remain immutable. Processes that run in the trusted path are described as being part of the Trusted Path Domain (TPD).

In immutable zones, certain core system processes are marked as part of the TPD. For example, a number of system daemons run in the TPD, including `init`, `svc.configd`, and `svc.startd`. When you are given administrative access to TPD processes, you can safely modify the configuration of an immutable zone because all non-TPD processes remain unwritable.

You can administer an immutable zone by using the trusted path locally through the console or remotely through a trusted `rad` connection.

- To enable local administration, you must ensure that the console is accessible through the ILOM, a serial connection, or through the graphical console. You enter the TPD by logging in on a console that is protected by the trusted path where you are also trusted.

For the procedure, see [How to Enable Administrative Access to an Immutable Zone From the Console](#).

- To enable remote administration through the Remote Administration Daemon (RAD), you must protect the RAD process with the trusted path and you must also be trusted.

For the procedure, see [How to Enable Remote Administrative Access to an Immutable Zone by Using RAD](#).

How to Enable Administrative Access to an Immutable Zone From the Console

Perform this task to leave the zone immutable and enable the administrator to access processes and files in the TPD from the console.

1. Assume the root role.

For more information, see [Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*](#).

2. Restrict access to the console by configuring the `tpdlogin` PAM module in the global zone.

For instructions, see [How to Restrict Access to the Trusted Path Domain in *Managing Authentication in Oracle Solaris 11.4*](#).

3. Modify the console login SMF service to run in the TPD.

```
# svccfg -s console-login:default
svc:/system/console-login:default> setprop start/trusted_path = true
svc:/system/console-login:default> refresh
svc:/system/console-login:default> exit
```

4. (Optional) Verify that the `trusted_path` attribute is set in the console.

```
# svcprop -p start/trusted_path console-login:default
true
```

▲ Caution:

When you set the `trusted_path` attribute in the console, you must restrict access to the console in the `/etc/security/tpdusers` file to prevent login by unauthorized users. You should have prevented unauthorized logins in [Step 2](#).

5. Restart the console login service.

```
# svcadm restart console-login:default
```

6. Log in to the immutable zone as one of the users in `/etc/security/tpdusers`.

- Log in to the console and answer the Trusted Path login prompt.
- On a physical console, invoke the Trusted Path login prompt by typing the secure attention key sequence:

- **Stop-A** (SPARC)
- **F1-A** (x86)

After login, you can administer files and processes that are in the TPD. You can also assume a role and administer the immutable zone in that role.

How to Enable Remote Administrative Access to an Immutable Zone by Using RAD

Perform this task to enable access to the immutable zone by using the Remote Access Daemon (RAD). The `rad:remote` SMF service is enabled to perform TPD authentication over RAD connections.

1. Assume the root role.

For more information, see [Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*](#).

2. Modify the `rad:remote` SMF service to run in the TPD.

```
# svccfg -s rad:remote
svc:/system/rad:remote> setprop method_context/trusted_path = true
svc:/system/rad:remote> refresh
svc:/system/rad:remote> exit
```

3. Verify that the `trusted_path` attribute is set for the `rad:remote` service.

```
# svcprop -p method_context/trusted_path rad:remote
true
```

4. Restart the remote RAD service.

```
# svcadm restart rad:remote
```

5. Enable one or more administrators to access the TPD over a RAD connection.

```
$ usermod -K tpd=yes username
```

These administrators can now log in to the immutable zone remotely over RAD.

See Also

For information about RAD access, see [Chapter 3, Managing User Accounts Interactively in *Managing User Accounts and User Environments in Oracle Solaris 11.4*](#).

To add an SMF service to run in the TPD, see [Adding the Puppet Service to the Trusted Path](#).

Administering an Immutable Zone by Making It Writable

The `zoneadm boot` subcommand provides two options that enable the global zone administrator to manually boot an immutable zone with either a writable root file system or with a transient writable root file system. The zone is in writable mode only until the next reboot.

-w

Manually boot the zone with a writable `root` file system.

-W

Manually boot the zone with a transient writable `root` file system. The system is rebooted automatically when the `self-assembly-complete` milestone is reached. The reboot places the zone under control of the MWAC policy again. This option is permitted when the zone has an MWAC policy of `none`.

Both the `-W` and `-w` options are ignored for zones that are not immutable zones.

The `zlogin` command provides two options for actions such as editing an immutable file or adding a new package. Use of these options require the `solaris.zone.manage/zonename` authorization.

-T

Enters an immutable zone with the trusted path attribute `PRIV_PROC_TPD` set. This session can modify files in the zone that are normally immutable. The session cannot read unprotected files.

-U

Performs the same process as the `-T` option, but in unsafe mode. In unsafe mode, unprotected files can be modified. You use this option for zones with the `flexible-configuration` MWAC security policy.



Note:

These options cannot be used with console login and are ignored for zones that are not immutable zones.

Immutable Global Zones

Immutable global zones extend immutable non-global zones to global zones.

Configuring an Immutable Global Zone

To configure an immutable global zone is similar to configuring an immutable non-global zone. The MWAC security policy is set with the `zonecfg` command, as [Setting the MWAC Security Policy](#) describes. After committing the zone configuration, the boot information is written and the boot archive is updated. The global zone becomes immutable immediately. No reboot is necessary.

The following information is specific to immutable global zones:

- If the global zone uses DHCP to set network interfaces, the `flexible-configuration` MWAC policy must be selected.
- The `rpool` dataset is restricted.

You can add an unrestricted sub-dataset by using the `zonecfg add dataset` command. An immutable global zone can only run zones in unrestricted datasets. All the children of an unrestricted dataset are also unrestricted.

Maintaining an Immutable Global Zone

The most secure method of maintaining the global zone is by using the trusted path. Trusted path is only available on the console, so ensure that the console is accessible through the ILOM, a serial connection or through the graphical console.

After a system is configured to be immutable, configure the console login with the trusted path. For the procedure, see [How to Enable Administrative Access to an Immutable Zone From the Console](#). After you have configured the console login, the `root` account cannot log in and administer the zone. You must log in as a user who is authorized to use the trusted path. After logging in, you can then assume a role.

When you run the `pkg update` command in an immutable global zone, the first boot is read-write. The system needs these permissions to perform the required self-assembly steps. When the self-assembly steps have been performed, the system becomes immutable again.

12

Troubleshooting Miscellaneous Oracle Solaris Zones Problems

This chapter contains the following zones troubleshooting information:

- [Installation Fails Due to Unmatched Allowable Packages](#)
- [Exclusive-IP Zone Is Using Device, so dladm reset-linkprop Fails](#)
- [zonecfg -r Fails](#)
- [Troubleshooting a Zone That Has Failed](#)
- [Insufficient Privileges Specified for the Zone](#)
- [Zone Does Not Halt](#)
- [Zone Access Denied](#)

Installation Fails Due to Unmatched Allowable Packages

The following error message when you attempt to install a `solaris` zone indicates a publisher mismatch between the global zone and the non-global zone:

```
The following pattern(s) did not match any allowable packages. Try
using a different matching pattern, or refreshing publisher information:
```

The global zone currently might be configured with a publisher that does not contain the same version of the installed system software. To install a non-global zone, the repository that you set as the `solaris` publisher origin must contain at least the same system software that is installed in the global zone where you are installing the non-global zone.

See [Non-Global Zone Cannot Be Installed in Updating Systems and Adding Software in Oracle Solaris 11.4](#) for more information.

Troubleshooting Installation

The following are recovery instructions for zone installation failures:

- If a zone installation is interrupted or fails, the zone is left in the `incomplete` state. Use the `zoneadm uninstall -F` command to reset the zone to the `configured` state.
- If an error message is displayed and the zone fails to install, perform the following steps.

1. Check the state of the zone.

Output is similar to the following:

```
global$ zoneadm list -cv
  ID NAME           STATUS   PATH                               BRAND  IP
  0 global          running /                               solaris shared
- zonename
configured /system/zones/zonename  solaris excl
```

2. If the state is listed as `incomplete`, uninstall the zone.

```
global$ pfexec zoneadm -z zonename uninstall
```

3. Make the corrections specified in the error message.

4. Rerun the `zoneadm install` command to install the zone.

- If a storage object contains any preexisting partitions, zpools, or UFS file systems, the `install` operation fails and an error message is displayed. Perform the following steps.

1. Put the zone in the `uninstalled` state.

```
global$ pfexec zoneadm -z zonename uninstall
```

2. Continue the installation and overwrite any preexisting data.

Use one of the following forms of the `-x` option in the `zoneadm -z zonename install` command. The `-x` option is similar to the `zpool create -f` command.

```
-x force-zpool-import
-x force-zpool-create= zpoolname [...]
-x force-zpool-create=zpoolname1, zpoolname2, zpoolname3
-x force-zpool-create-all
```

For more information about using the `-x` option, see the [zoneadm\(8\)](#) man page.

Exclusive-IP Zone Is Using Device, so `dladm reset-linkprop` Fails

The following error message indicates that an attempt to use the `dladm reset-linkprop` command failed.

```
dladm: warning: cannot reset link property 'zone' on 'net0': operation failed
```

See [Managing Datalinks Related to Zones](#). The running zone `excl` is using the device.

To reset the value, do the following:

1. Type the following command:

```
global$ ipadm delete-ip net0
```

2. Rerun the `dladm` command.

`zonecfg -r` Fails

The following error message indicates that an attempt to configure a zone in live mode failed:

```
Failed to commit. the live configuration of the zone changed externally.
```

To enable the `commit` to succeed, refresh the zone snapshot with the running zone configuration:

```
zonecfg:zonename> reload
```

Repeat the configuration changes that generated the error message.


```
zonecfg:zonename> zonecfg-subcommands
zonecfg:zonename> commit
zonename: Checking: zonecfg-subcommands
zonename: Applying changes
```

Troubleshooting a Zone That Has Failed

The unexpected or unplanned unavailability of zones can have multiple causes. Whatever the causes, the resulting problem can be simplified to one of two kinds: a temporary problem or a permanent one. The following procedure describes basic steps to troubleshoot and possibly fix a zone that has become inaccessible.

About Zone UUIDs

Zones can be uninstalled and reinstalled under the same name with different contents. Zones can also be renamed without the contents being changed. For these reasons, the UUID is more reliable than the zone name, especially when you need to troubleshoot problems with zones.

To obtain the UUID of a zone, use the `-p` option when listing zones:

```
global$ zoneadm list -cp
...
ID:zonename:status:zonepath:UUID
```

Thereafter, you can use `-u UUID` option instead of the more typical `-z zonename` option when you run `zoneadm` or `zonecfg` commands.

You can use both the `-z zonename` and `-u UUID` options together in the command. In this case, the command attempts to match the UUID first. If a match is found, then the `-z zonename` option is ignored. Otherwise, the command uses the zone name.

Tip:

UUIDs are long and can potentially cause problems when used in commands. Obtain the UUID of a problematic zone first to identify the corresponding zone name. Then, proceed with using the zone name for convenience.

How to Manage a Zone That Has Become Unavailable

For demonstration purposes, this procedure refers to the UUID of the zone instead of the zone name.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Display the UUIDs of the zones on the system.

```
global$ zoneadm list -cp
0:global:running://:solaris:shared:-:
ID:zonename:status:zonepath:UUID
```

3. If necessary or possible, shut down the zone.

```
global$ zoneadm -u UUID shutdown
```

4. Verify that the zone is in the installed state.

```
global$ zoneadm list -cp
```

An `installed` status indicates that a zone is not booted.

5. Determine whether the zone's unavailability is temporary or permanent.

Your investigation and analysis might extend beyond just zone-specific issues.

6. Change the status of the zone.

The status depends on the nature of the problem.

- The problem is temporary.

Temporary failures might be caused by dropped network connections, misconfiguration, and other resolvable issues. For this type of problem, do the following:

- a. Mark the zone's status as unavailable.

```
global$ zoneadm -u UUID mark unavailable
```

The `unavailable` status indicates that an installed zone cannot be booted. This state is displayed by the `zoneadm list -p` command.

- b. Fix the problem that caused the zone to fail.
- c. After the problem is resolved, return the zone to the `installed` state.

```
global$ zoneadm -u UUID attach
```

- d. (Optional) Put the zone in a ready state.

```
global$ zoneadm -u UUID mark ready
```

Perform this step if you do not want to use the zone at this time. Booting the zone in the next step automatically brings the zone to a ready state before booting.

- e. Boot the zone.

```
global$ zoneadm -u UUID boot
```

- The problem is permanent.

Permanent problems might be in the form of corrupt ZFS file systems, for example. You cannot recover a zone that undergoes a permanent failure.

- a. Mark the zone's status as incomplete.

```
global$ zoneadm -u UUID mark incomplete
```

 **Note:**

Marking a zone `incomplete` is irreversible.

- b. Uninstall and remove the zone.

See [How to Uninstall and Remove a Zone](#).

Insufficient Privileges Specified for the Zone

If the zone's privilege set contains a disallowed privilege, is missing a required privilege, or includes an unknown privilege name, an attempt to verify, ready, or boot the zone will fail with an error message such as the following:

```
zonecfg:zone5>set limitpriv="basic"
global$ zoneadm -z zone5 boot
required privilege "sys_mount" is missing from the zone's privilege set
zoneadm: zone zone5 failed to verify
```

To fix the problem, add to the zone's `limitpriv` property. The following example shows how to add the DTrace privileges to a zone.

Example 12-1 Adding DTrace Privileges to a Non-Global Zone

In this example, the administrator adds DTrace privileges to an installed non-global zone that is not booted. When added to a zone, DTrace privileges enable the use of the DTrace facility. Users with the DTrace Toolkit rights profile as well as the appropriate Zones rights profiles will be able to use DTrace functionality as described in [Running DTrace in a Non-Global Zone](#).

1. From the global zone, the administrator adds the `dtrace_proc` and `dtrace_user` privileges to the non-global zone named `dev-zone`.

```
global# zonecfg -z dev-zone
zonecfg:dev-zone> set limitpriv="default,dtrace_proc,dtrace_user"
zonecfg:dev-zone> exit
```

2. The administrator then boots the zone, logs in as `developer-1` who has been assigned the DTrace Toolkit rights profile only, and tests that `developer-1` can use the `dtrace` command.

```
global# zoneadm -z zonename boot

# su developer-1
developer-1$ zlogin dev-zone

dev-zone: developer-1$ dtrace -l
```

For more information, see the [zonecfg\(8\)](#) man page and [limitpriv Global Property in Oracle Solaris Zones Configuration Resources](#).

Zone Does Not Halt

If the system state associated with the zone cannot be destroyed, the halt operation will fail halfway. This leaves the zone in an intermediate state, somewhere between `running` and `installed`. In this state there are no active user processes or kernel threads, and none can be created. When the halt operation fails, you must manually intervene to complete the process.

The most common cause of a failure is the inability of the system to unmount all file systems. Unlike a traditional Oracle Solaris system shutdown, which destroys the system state, zones must ensure that no mounts that were performed while booting the zone or during zone operation remain once the zone has been halted. Even though the `zoneadm` command makes sure that there are no processes executing in the zone, the unmount operation can fail if processes in the global zone have open files in the zone.

Use the tools described in the `proc(1)` man page (see the `pfiles(1)` man page) and the `fuser(8)` man page to find these processes and take appropriate action. After these processes are dealt with, rerun the `zoneadm halt` command to completely halt the zone.

Zone Access Denied

When a connection to the zone is denied, use the `zlogin -S` command to enter the zone in Failsafe mode.

Oracle Solaris Zones on Shared Storage

You can use Oracle Solaris Zones on shared storage (ZOSS) to transparently access and manage shared storage resources in zones. These automated capabilities simplify deployment, administration, and migration of zones and their corresponding shared storage resources in Oracle Solaris systems.

You can describe the corresponding shared storage resources in a host-independent format in the zone configuration. Zones installations using this feature are encapsulated into dedicated ZFS storage pools hosted on shared storage devices.

The Oracle Solaris Zones framework will automatically configure and unconfigure shared storage resources. Any ZFS storage pool management tasks required throughout the various zone management activities will be performed automatically.

This chapter provides the following information:

- [About Shared Storage Resources Using Storage URIs](#)
- [Managing Storage URIs and Shared Storage Resources](#)
- [Assigning Shared Storage Resources to Oracle Solaris Zones](#)
- [Automated ZFS Storage Pool Management for Oracle Solaris Zones on Shared Storage Resources](#)
- [About the unavailable Zone State on Shared Storage](#)
- [zoneadm Options for Shared Storage](#)
- [Implementing Oracle Solaris Zones Hosted on Shared Storage Resources](#)
- [Migrating Oracle Solaris Zones Hosted on Shared Storage Resources](#)
- [Moving Existing Zones To or From Shared Storage Zone Configurations](#)
- [Zones Documentation References for Shared Storage Resources](#)

About Shared Storage Resources Using Storage URIs

Storage Universal Resource Identifiers (URIs) are used to describe shared storage resources in a host-independent format. Storage URIs uniquely identify storage objects across different nodes. The URIs follow the well-known principles and format of URIs commonly used in the Internet. The following storage URIs are available.

Local Device URI

The local device storage URI type describes a storage device by its local device path. The path must refer to a device in the `/dev` name space. These devices are usually direct-attached storage (DAS) resources that are unique to a particular system, and their device name and path are generally not portable. However, they can also refer to shared storage resources for technologies that already provide a unified name space under `/dev` across multiple nodes.

The following is a list of URI syntax:

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

The following examples show how to use the URI syntax:

```
dev:dsk/c0t0d0s0
dev:///dev/dsk/c0t0d0
dev:/dev/dsk/c0t0d0
dev:chassis/SYS/HD1/disk
dev:dsk/c0t60A98000564C303132302D6F72613939d0
```

The local device storage URI can refer to an entire disk or to a particular slice or partition. However, the use of slices or partitions is generally not recommended with ZFS storage pools.

Logical Unit URI

The logical unit URI type describes fibre channel (FC) or serial-attached SCSI (SAS) storage devices. It refers to a logical unit (LU) based on its device ID (WWN). The logical unit storage URI always represents an entire disk.

The following is a list of URI syntax:

```
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
```

The following examples show how to use the URI syntax:

```
lu:luname.naa.5000c5000288fa25
lu:luname.eui.0021280001cf80f6
lu:initiator.naa.2100001d38089fb0,target.naa.2100001d38089fb0,luname.naa.5000c500
0288fa25
lu:initiator.naa.2100001d38089fb0,target.naa.2100001d38089fb0,luname.eui.
0021280001cf80f6
```

In the `luname-only` URI form, the ID describes a logical unit name.

In the `initiator,target,luname` form, an initiator specifies an initiator port and a target specifies a target port. Together, they specify a path to the logical unit. The logical unit name in the second syntax must match the URI's `luname` in the first syntax, the `luname-only` form.

It is best to use multipathing in conjunction with `luname-only` URIs. If multipathing is disabled and a `luname-only` URI is used, a random path to the specified logical unit will be chosen. To avoid random path assignment an `initiator,target,luname` URI form can be used to select a specific path to a logical unit. If multipathing is enabled and an `initiator,target,luname` URI form is used, then the multipathing framework controls which paths are used to access the logical unit, and the URI is only used to identify the unit, not the access path.

iSCSI URI

The iSCSI URI type describes storage devices accessed by using the iSCSI network-based storage protocol. It always refers to an entire disk.

For more information about iSCSI Qualified Name (IQN), see the [suri\(7\)](#) man page.

The following is a list of URI syntax:

```
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
```

The following examples show how to use the URI syntax:

```
iscsi:///lunname.eui.0021280001cf80f6
iscsi:///lunname.naa.600144f03d70c80000004ea57da10001
iscsi://[::1]/lunname.naa.600144f03d70c80000004ea57da10001
iscsi://127.0.0.1/lunname.naa.600144f03d70c80000004ea57da10001
iscsi://127.0.0.1:3260/lunname.naa.600144f03d70c80000004ea57da10001
iscsi://hostname:3260/lunname.eui.0021280001cf80f6
iscsi://hostname:3260/lunname.naa.600144f03d70c80000004ea57da10001
iscsi://[::1]/target.iqn.com.sun:02:d0f2d311-f703,lun.2
```

See [Logical Unit URI](#) for an explanation of the luname-only URI form and the ID.

Optionally, the `hostname[:port]` authority section provides information to automatically configure the iSCSI initiator using an unauthenticated `SendTargets` discovery address. IPv6 addresses must be enclosed in square brackets (`[]`).

Use the luname-only URI form only for static or iSNS based configurations, or when using authentication. These functions must be configured outside of the zones framework before they can be used.

Note that if you want to use iSCSI over iSER transports, the iSCSI storage URI transparently allows for this as well. To use iSER, the target and initiator must go through the address assigned to the InfiniBand (IB) partition link. In the global zone, you must set up the IB partitions with the `dladm create-part` command and assign the target and initiator addresses to the IB partitions using the `ipadm create-ip` and `ipadm create-addr` commands. This address is then used in the authority section of the iSCSI storage URI to specify the target's discovery address.

If InfiniBand (IB) hardware is present and an InfiniBand reliable-connected (RC) connection can be established, then an iSER-enabled initiator uses iSER connections to iSER-enabled targets. If the RC connection cannot be established, the connection is established using IP-based connectivity.

Managing Storage URIs and Shared Storage Resources

To generate and verify storage URIs or administer shared storage resources based on storage URIs, use the `suriadm` command.

You can use the `suriadm` command to verify storage URIs that were created manually, or to create storage URIs automatically, based on existing system device paths. Depending on the

storage URI type, the `suriadm` command allows you to configure and unconfigure the corresponding storage subsystem. Given a storage URI, the same command can also identify possible instantiated device instances for the storage object described by the storage URI. The following examples demonstrate common use cases. For more information, see the `suriadm(8)` man page.

Example 13-1 Verifying Storage URIs With `suriadm parse`

The following examples show how to use the `suriadm` command to verify storage URIs. In these examples, the storage URI string has been parsed and its corresponding properties are displayed. Once verified, the storage URI can be used later with either the `zonecfg` command or the `suriadm` command.

```
root@initiator:~# suriadm parse iscsi://target/
lunname.naa.600144F035FF850000050C884E50001
PROPERTY      VALUE
uri-type      iscsi
hostname      target
port          -
lunname       naa.600144F035FF850000050C884E50001

root@host:~# suriadm parse dev:/dev/dsk/c4t1d0
PROPERTY      VALUE
uri-type      dev
path          /dev/dsk/c4t1d0
```

Example 13-2 Producing Storage URIs Based on Device Path With `suriadm lookup-uri`

The following example shows how to use the `suriadm` command to produce storage URIs. In the example, based on the existing local device path, the `suriadm` command output suggests valid storage URIs for later use with either the `suriadm` or `zonecfg` commands.

```
root@target:~# suriadm lookup-uri -t iscsi /dev/dsk/
c0t600144F035FF850000050C884E50001d0
iscsi://target/lunname.naa.600144f035ff850000050c884e50001

root@host:~# suriadm lookup-uri /dev/dsk/c4t1d0
dev:dsk/c4t1d0

root@host:~# suriadm lookup-uri /dev/dsk/c0t600144F0DBF8AF190000510979640005d0
lu:lunname.naa.600144f0dbf8af190000510979640005
lu:initiator.naa.10000000c9991d8c,target.naa.21000024ff3ee89f,lunname.naa.600144f0dbf8af190000510979640005
dev:dsk/c0t600144F0DBF8AF190000510979640005d0
```

Example 13-3 Configuring iSCSI based Storage Resources With `suriadm map`

The following example shows how to use the `suriadm map` command to configure iSCSI based storage resources. In this example, for an iSCSI storage URI, the `suriadm` configures the iSCSI initiator's send-targets discovery address and instantiates a local device representing the iSCSI target. The local device path from the `mapped-dev` storage URI property can now be used with utilities such as the `zpool`, `format`, and `mkfs` commands.

```
root@initiator:~# suriadm map iscsi://target/
lunname.naa.600144F035FF850000050C884E50001
```



```
PROPERTY      VALUE
mapped-dev    /dev/dsk/c0t600144F035FF8500000050C884E50001d0s0
```

Example 13-4 Locating a Configured Storage Resource With `suriadm lookup-mapping`

In this example, the command is used to show the local system device currently associated with the given storage URI.

```
root@initiator:~# suriadm lookup-mapping iscsi://target/
lunname.naa.600144F035FF8500000050C884E50001
PROPERTY      VALUE
mapped-dev    /dev/dsk/c0t600144F035FF8500000050C884E50001d0s0
```

Example 13-5 Unconfiguring iSCSI-Based Storage Resources With `suriadm unmap`

For an iSCSI storage URI, the `suriadm` command removes the iSCSI initiator's send-targets discovery address, and unconfigures the shared storage resource.

```
root@initiator:~# suriadm unmap iscsi://target/
lunname.naa.600144F035FF8500000050C884E50001
root@initiator:~# suriadm lookup-mapping iscsi://target/
lunname.naa.600144F035FF8500000050C884E50001
Failed to lookup mapping for URI: "iscsi://target/
lunname.naa.600144F035FF8500000050C884E50001": No such logical
unit name found: "naa.600144F035FF8500000050C884E50001"
```

Assigning Shared Storage Resources to Oracle Solaris Zones

Assign shared storage resources to zones in the zone configuration by using storage URIs to describe the location of a storage object.

Two zone configuration resource types, `rootzpool`, and `zpool`, and a property type, `storage`, are used to assign shared storage resources to a particular Oracle Solaris Zone. These are configured and maintained with the `zonecfg`.

storage Property for Zones

When using shared storage resources, the `storage` property defines the location of the storage object in a host-independent format by using storage URIs. The following storage URIs are currently supported by the Oracle Solaris Zones framework in Oracle Solaris:

- `dev`: local device path storage URI, DAS
- `iscsi`: iSCSI storage URI
- `lu`: Fibre Channel (FC) and Serial Attached SCSI (SAS)

The `storage` property is managed using the following `zonecfg` subcommands from within a `rootzpool` or `zpool` resource scope:

```
zonecfg:zonename:zpool> add storage URI-string
zonecfg:zonename:zpool> remove storage URI-string
```

rootzpool Resource

The `rootzpool` resource is a dedicated ZFS storage pool for a zone. The entire zone installation is encapsulated into its own dedicated ZFS storage pool. This ZFS storage pool will be composed of shared storage resources.

The `rootzpool` resource must specify at least one storage property. Multiple storage properties can be specified to describe redundant ZFS storage pool configurations. Note that there can be only one `rootzpool` resource per zone configuration.

The ZFS storage pool name for a `rootzpool` resource will be automatically assigned as `zonename_rpool`. The name cannot be changed. The `rootzpool` resource is managed by using the following `zonecfg` subcommands from the global resource scope:

```
zonecfg:zonename> add rootzpool
zonecfg:zonename:rootzpool> add storage URI-string
zonecfg:zonename:rootzpool> end

zonecfg:zonename> remove rootzpool

zonecfg:zonename> select rootzpool storage=URI-string

zonecfg:zonename> info rootzpool
```

zpool Resource

The `zpool` resource describes a ZFS storage pool composed of shared storage resources that is delegated to the Oracle Solaris Zone. The `zpool` resource will specify at least one storage property. Multiple storage properties can be specified to describe redundant ZFS storage pool configurations. There can be multiple `zpool` resources defined for a zone configuration.

The ZFS storage pool name for a `zpool` resource is assigned by combining the zone name and the specified name property, as `zonename_name`. For the name property, the `zonecfg` will verify that the string is eligible for a ZFS storage name and a ZFS dataset name. The string `rpool` is not permitted and cannot be used for this property.

The `zpool` resource is managed using the following `zonecfg` subcommands from the global resource scope:

```
zonecfg:zonename> add zpool
zonecfg:zonename:zpool> add storage URI-string
zonecfg:zonename:zpool> set name=name-string
zonecfg:zonename:zpool> end

zonecfg:zonename> remove zpool
zonecfg:zonename> remove zpool name=name-string
zonecfg:zonename> remove zpool storage=URI-string

zonecfg:zonename> select zpool storage=URI-string

zonecfg:zonename> info zpool
zonecfg:zonename> info zpool name=name-string
zonecfg:zonename> info zpool storage=URI-string
```

Renaming Zones

You can use the `zoneadm` command to rename a zone that is in either the `configured` or the `installed` state.

```
global$ zoneadm -z zonename rename new-zonename
```

In addition, you can use the `zonecfg` command to rename a zone that is in the `configured` state.

```
zonecfg:zonename> set zonename=new-zonename
```

Relocating Shared Storage Resource

To alter a storage resource URI within a `rootzpool` or `zpool` resource when the location description changes for an existing shared storage resource, use the `zonecfg remove storage old-URI` command syntax followed by the `add storage new-URI` command.

Automated ZFS Storage Pool Management for Oracle Solaris Zones on Shared Storage Resources

When using the zones on the shared storage support feature in Oracle Solaris, the zones framework will automatically manage all of the ZFS storage pools associated with the `rootzpool` or `zpool` resources for a particular zone.

To facilitate zone and storage resource migration, ZFS storage pools associated with `rootzpool` or `zpool` resources are not persistently configured on the system. They will not appear in the system's global ZFS storage pool repository `/etc/zfs/zpool.cache`.

The zones framework creates corresponding ZFS storage pools when you install or clone a zone. You can configure more than one shared storage resource with the `zonecfg add storage` command per `rootzpool` or `zpool` resource. In this case, a mirrored ZFS storage pool will be created by default.

You can create ZFS storage pools in advance by using custom configurations such as encryption), or different redundancy levels like `raidz` or `raidz2`. After first configuring all the required shared storage resources in the zone configuration, use the `zpool` utility to export the pre-created ZFS storage pool again. The zones framework will first attempt to import and use this pre-created ZFS storage pool during zone installation and cloning.

When you use the `zoneadm attach` syntax, the zones framework will initially configure all shared storage resources and then import all configured ZFS storage pools.

For the `zoneadm detach` command syntax, the zones framework will first export all configured ZFS storage pools and then unconfigure all shared storage resources.

When uninstalling a zone with the `zoneadm uninstall` command, the zones framework will first export all configured ZFS storage pools and then unconfigure all shared storage resources by default. However, you can explicitly request that ZFS storage pools be destroyed by using the `-x force-zpool-destroy` option with the `zoneadm uninstall` command.

During system boot, the zones SMF service `svc:/system/zones:default` is run. The SMF service will configure shared storage resources and import ZFS storage pools for all zones in the `installed` state that have `rootzpool` or `zpool` resources. Any failures during this stage will be logged to the corresponding SMF services log file `/var/svc/log/system-zones:default.log`.

ZFS storage pools configured with `rootzpool` or `zpool` resources will appear as virtualized ZFS storage pools inside the zone, but they cannot be managed directly by the zone administrator.

While in use by a running zone, ZFS storage pools cannot be destroyed or exported by the global zone.

From the global zone, you can use the `zpool` command for administrative actions on ZFS storage pools managed by the zones framework. Administrative actions include bringing a specific device online or taking it offline, replacing faulted devices, and adding or removing devices. To keep the zone configuration in sync, update the corresponding storage URIs accordingly to reflect changes made to the shared storage resources being used.

About the `unavailable` Zone State on Shared Storage

In Oracle Solaris, the zone state `unavailable` indicates that the zone is installed but cannot be booted. This state is displayed in the output of the `zoneadm list -p` command syntax.

```
root@initiator:~# zoneadm list -cp
0:global:running:/::solaris:shared:-:none
-:iscsi:unavailable:/iscsi:a0a4ba0d-9d6d-cf2c-cc42-f123a5e3ee11:solaris:excl:-:
```

An installed zone with shared storage resources can enter the `unavailable` state if the zones framework experiences either failures during the configuration of shared storage resources or failures during ZFS storage pool management.

To move the zone out of the `unavailable` state, you must first identify and possibly fix any problems related to shared storage resource connectivity or zone misconfiguration. You can then use the `zoneadm attach` command to reattach the zone properly and move it into the `installed` state. It is also possible to uninstall a zone with the `zoneadm uninstall` command to move the zone back into the `configured` state.

For more information about Zones states, see [Non-Global Zone State Model in Introduction to Oracle Solaris Zones](#).

`zoneadm` Options for Shared Storage

Where appropriate, the `zoneadm` CLI subcommands have been enhanced with options specific to ZFS storage pool management.

Options for Installing, Cloning, and Attaching Zones

The `install`, `clone`, and `attach` subcommands of `zoneadm` have the following options:

- `-x force-zpool-import` Option

- [-x force-zpool-create= Option](#)
- [-x force-zpool-create-all Option](#)

-x force-zpool-import Option

This option applies to all `zpool` resources specified in the zone configuration and instructs the zones framework to forcibly import any ZFS storage pools that might appear to be in use, such as by another system. This mimics the behavior of `zpool import -f` command.

This option facilitates importing ZFS storage pools onto a new system during zone migration when the ZFS storage pools were never properly exported on another system, for example, by using the `zoneadm detach` or `zoneadm uninstall` commands. In these situations, the forcible import must be applied to all ZFS storage pools configured for a particular zone.

-x force-zpool-create= Option

This option instructs the zones framework to forcibly create a new ZFS storage pool, either over an existing old pool or onto a device that appears to be currently in use for other purposes. This mimics the behavior of the `zpool create -f` command syntax.

This option's scope is limited to the ZFS storage pools specified in a comma-separated list of `zpool` names or specified by using the option multiple times, for example:

```
-x force-zpool-create=rpool,pool2,pool3
-x force-zpool-create=pool1 -x force-zpool-create=pool2
```

The ZFS storage pool name to be used here is the name property of the corresponding `zpool` resource in the zone configuration. To specify the ZFS storage pool described by the `rootzpool` resource, use the name `rpool`.

-x force-zpool-create-all Option

This option instructs the zones framework to forcibly create new ZFS storage pools for all `zpool` and `rootzpool` resources specified in the zone configuration.

Options for Uninstalling Zones

The `zoneadm uninstall` subcommand has the following options:

- [-x force-zpool-destroy=zpoolname Option](#)
- [-x force-zpool-destroy-all Option](#)
- [-x force-storage-destroy-all Option](#)

-x force-zpool-destroy=zpoolname Option

This option instructs the zones framework to destroy a ZFS storage pool while uninstalling a zone. This mimics the behavior of `zpool destroy poolname` command. This option takes a `zpoolname` list:

```
force-zpool-destroy=zpoolname{,zpoolname,zpoolname,...}
```

The scope of this option is limited to the ZFS storage pools specified in the comma-separated list of `zpool` names, or specified by using the option multiple times, for example:

```
-x force-zpool-destroy=rpool,pool2,pool3  
-x force-zpool-destroy=pool1 -x force-zpool-destroy=pool2
```

The name arguments correspond to the name property specified in the zone configuration for the desired `zpool` resource. The name `rpool` is used to specify the ZFS storage pool associated with the `rootzpool` resource.

-x force-zpool-destroy-all Option

This option instructs the zones framework to destroy the ZFS storage pools of all `zpool` and `rootzpool` resources specified in the zone configuration when uninstalling the zone.

Note that the options to force the zones framework to destroy ZFS storage pools when uninstalling a zone should be used with caution. Even though you might want to uninstall a zone in the current, active boot environment (BE), there could be other zone boot environments (ZBEs) within this ZFS storage pool that belong to currently inactive boot environments. Destroying the ZFS storage pool associated with a `rootzpool` resource will make the zone unavailable to all boot environments that carry this zone in the `installed` state. For more information about zone boot environments, refer to the [beadm\(8\)](#) man page.

-x force-storage-destroy-all Option

Specify this option to destroy storage. Note that not all storage URI types support this operation. You can destroy storage referenced by file or NFS storage URIs, but not storage referenced by `dev`, `iSCSI`, or `lu` storage URIs.

Restrictions on Moving a Zone With the `rootzpool` Resource

For zones that are configured with a `rootzpool` resource, you cannot use the `zoneadm move` command to change the path name of the ZFS storage pool. However, you can use this command to change the underlying storage devices of the ZFS storage pool.

Implementing Oracle Solaris Zones Hosted on Shared Storage Resources

The following sections discuss deploying Oracle Solaris Zones on shared storage resources.

`zpool` Configuration Considerations for Zones on Shared Storage

When configuring ZFS storage pools, you might want to consider using redundant pool configurations such as `mirror`, `raidz`, or `raidz2`. Even if the backend storage hosting the shared storage resources provides resilience and data protection, in the form of hardware RAID for example, redundant configurations enable ZFS storage pools to use their inherent self-healing capabilities.

There are two ways to use redundant ZFS storage pool configurations with Oracle Solaris Zones hosted on shared storage resources.

- You can use mirrored ZFS storage pool configurations, which by default will be created automatically if you specify multiple storage properties for a `rootzpool` or `zpool` resource in the zone configuration.
- Alternatively, you can create more complex ZFS storage pool configurations such as `raidz` or `raidz2` in advance, with the `zpool create` command. Note that you must export the new created ZFS storage pool again using `zpool export`. Next, all corresponding storage resource URIs should be added to the `rootzpool` or `zpool` resource in the zone configuration with the `zonecfg` command. The zones framework will now import this ZFS storage pool during zone installation or clone rather than attempt to create a new ZFS storage pool.

Sample Scenarios for Zones on Shared Storage

This section provides examples for deploying Oracle Solaris Zones on shared storage resources.

Example 13-6 Using iSCSI-Based Shared Storage Devices With Oracle Solaris Zones

This exercise sets up a sample configuration on an Oracle Solaris server that will provide shared storage through an iSCSI target. A zone is configured and installed on a second server running Oracle Solaris, using the iSCSI-based shared storage resources to host a zone.

First, install the corresponding package, using one of the following `pkg install` commands. The first command installs the entire multi-protocol storage-server group package. The second command installs only the target support for iSCSI within the common multi-protocol SCSI target (COMSTAR) framework, as described in the [itadm\(8\)](#) and [stmfadm\(8\)](#) man pages.

```
root@target:~# pkg install group/feature/storage-server
root@target:~# pkg install system/storage/iscsi/iscsi-target
```

Then, create the backing store for the iSCSI targets to be exported from this server. Create four ZFS volumes as the backing store for four iSCSI target logical units, each 10GB in size, stored in the target servers `rpool/export` dataset with the `zfs` command.

```
root@target:~# zfs create -V 10G rpool/export/zonevol1
root@target:~# zfs create -V 10G rpool/export/zonevol2
root@target:~# zfs create -V 10G rpool/export/zonevol3
```

After setting up the backing store, use the `stmfadm` command to create target logical units for each ZFS volume. This gives the corresponding device ID (WWN) for each, which will be used later in the storage URI for iSCSI target discovery on the client host.

```
root@target:~# stmfadm create-lu /dev/zvol/rdisk/rpool/export/zonevol1
Logical unit created: 600144F035FF8500000050C884E50001
root@target:~# stmfadm create-lu /dev/zvol/rdisk/rpool/export/zonevol2
Logical unit created: 600144F035FF8500000050C884E80002
root@target:~# stmfadm create-lu /dev/zvol/rdisk/rpool/export/zonevol3
Logical unit created: 600144F035FF8500000050C884EC0003
```

You can view configured logical units with the `stmfadm list-lu` syntax.

```
root@target:~# stmfadm list-lu
LU Name: 600144F035FF8500000050C884E50001
LU Name: 600144F035FF8500000050C884E80002
LU Name: 600144F035FF8500000050C884EC0003
LU Name: 600144F035FF8500000050C884EC0004
```

You can query for details about configured logical units with the `stmfadm list-lu -v` syntax.

```
root@target:~# stmfadm list-lu -v
LU Name: 600144F035FF8500000050C884E50001
  Operational Status      : Online
  Provider Name          : sbd
  Alias                  : /dev/zvol/rdisk/rpool/export/zonev011
  View Entry Count       : 0
  Data File              : /dev/zvol/rdisk/rpool/export/zonev011
  Meta File              : not set
  Size                   : 10737418240
  Block Size             : 512
  Management URL         : not set
  Software ID            : not set
  Vendor ID              : SUN
  Product ID             : COMSTAR
  Serial Num             : not set
  Write Protect          : Disabled
  Write Cache Mode Select: Enabled
  Writeback Cache        : Enabled
  Access State           : Active
```

To make the logical unit available to iSCSI initiators, add a logical unit view to the target server with the `stmfadm add-view` command.

```
root@target:~# stmfadm add-view 600144F035FF8500000050C884E50001
root@target:~# stmfadm add-view 600144F035FF8500000050C884E80002
root@target:~# stmfadm add-view 600144F035FF8500000050C884EC0003
```

Now configure the iSCSI target on the target server. First, enable the iSCSI target SMF service with `svcadm enable`.

```
root@target:~# svcadm enable -r svc:/network/iscsi/target:default
```

Then, create the iSCSI target itself using `itadm create-target`.

```
root@target:~# itadm create-target
Target iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a successfully
created
```

You can query for the details about configured iSCSI targets by using either `itadm list-target` or `stmfadm list-target`.

```
root@target:~# itadm list-target -v
TARGET NAME                                STATE    SESSIONS
iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a  online    0
  alias:                                    -
  auth:                                     none (defaults)
  targetchapuser:                           -
  targetchapsecret:                         unset
  tpg-tags:                                  default

root@target:~# stmfadm list-target -v
Target: iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a
  Operational Status      : Online
  Provider Name          : iscsit
  Alias                  : -
  Protocol                : iSCSI
  Sessions                : 0
```


This completes the tasks performed on the sample server that provides the iSCSI target storage.

The next steps are to configure and install a zone on the second server by using this shared storage provided over iSCSI.

The first step is to install the corresponding package on the client server selected to be the iSCSI initiator.

```
root@initiator:~# pkg install pkg:/system/storage/iscsi/iscsi-initiator
```

Next, use the `zonecfg` command to configure a zone with a `rootzpool` and a `zpool` resource. Use the three iSCSI target logical units configured as shared storage resources to host the zone. Use the iSCSI storage URIs obtained previously with `suriadm`, on the target server.

```
root@initiator:~# zonecfg -z iscsi
Use 'create' to begin configuring a new zone.
zonecfg:iscsi> create
create: Using system default template 'SYSdefault'
zonecfg:iscsi> add rootzpool
zonecfg:iscsi:rootzpool> add storage iscsi://target/
lunname.naa.600144F035FF8500000050C884E50001
zonecfg:iscsi:rootzpool> end
zonecfg:iscsi> add zpool
zonecfg:iscsi:zpool> set name=data
zonecfg:iscsi:zpool> add storage iscsi://target/
lunname.naa.600144F035FF8500000050C884E80002
zonecfg:iscsi:zpool> add storage iscsi://target/
lunname.naa.600144F035FF8500000050C884EC0003
zonecfg:iscsi:zpool> end
zonecfg:iscsi> commit
zonecfg:iscsi> exit
```

Install the zone using `zoneadm install`.

```
root@initiator:~# zoneadm -z iscsi install
Configured zone storage resource(s) from:
    iscsi://target/lunname.naa.600144F035FF8500000050C884E50001
Created zone zpool: iscsi_rpool
Configured zone storage resource(s) from:
    iscsi://target/lunname.naa.600144F035FF8500000050C884E80002
    iscsi://target/lunname.naa.600144F035FF8500000050C884EC0003
Created zone zpool: iscsi_data
Progress being logged to /var/log/zones/zoneadm.20130125T112209Z.iscsi.install
Image: Preparing at /iscsi/root.

AI Manifest: /tmp/manifest.xml.pmai7h
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
Zonename: iscsi
Installation: Starting ...

    Creating IPS image
Startup linked: 1/1 done
    Installing packages from:
        solaris
            origin: http://pkg.oracle.com/solaris/release/
DOWNLOAD          PKGS          FILES      XFER (MB)   SPEED
Completed          183/183    33556/33556  222.2/222.2  3.4M/s

PHASE              ITEMS
```

```

Installing new actions                46825/46825
Updating package state database       Done
Updating image state                  Done
Creating fast lookup database         Done
Installation: Succeeded

```

Note: Man pages can be obtained by installing `pkg:/system/manual` done.

Done: Installation completed in 266.487 seconds.

Next Steps: Boot the zone, then log into the zone console (`zlogin -C`)

to complete the configuration process.

```

Log saved in non-global zone as /iscsi/root/var/log/zones/
zoneadm.20130125T112209Z.iscsi.install
root@initiator:~#

```

With the zone installation completed, verify that the zone has been properly installed with `zoneadm list` command.

```

root@initiator:~# zoneadm list -cp
0:global:running:/::solaris:shared:-:none
-:iscsi:installed:/iscsi:a0a4ba0d-9d6d-cf2c-cc42-f123a5e3ee11:solaris:excl:-:

```

Observe the newly created ZFS storage pools associated with this zone by using the `zpool list` command.

```

root@initiator:~# zpool list
NAME          SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
iscsi_data    9.94G  83.5K  9.94G  0%   1.00x  ONLINE  -
iscsi_rpool   9.94G  436M   9.51G  4%   1.00x  ONLINE  -

```

```

root@initiator:~# zpool status -v iscsi_rpool
pool: iscsi_rpool
state: ONLINE
scan: none requested
config:

```

NAME	STATE	READ	WRITE	CKSUM
iscsi_rpool	ONLINE	0	0	0
c0t600144F035FF8500000050C884E50001d0	ONLINE	0	0	0

```

root@initiator:~# zpool status -v iscsi_data
pool: iscsi_data
state: ONLINE
scan: none requested
config:

```

NAME	STATE	READ	WRITE	CKSUM
iscsi_data	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c0t600144F035FF8500000050C884E80002d0	ONLINE	0	0	0
c0t600144F035FF8500000050C884EC0003d0	ONLINE	0	0	0

The zone installation will be entirely contained within this ZFS storage pool. The ZFS dataset layout for this zone follows.

```

root@initiator:~# zfs list -t all|grep iscsi
iscsi_data                83.5K  9.78G   31K  /iscsi_data
iscsi_rpool               436M  9.36G   32K  /iscsi
iscsi_rpool/rpool        436M  9.36G   31K  /rpool
iscsi_rpool/rpool/ROOT   436M  9.36G   31K  legacy
iscsi_rpool/rpool/ROOT/solaris 436M  9.36G  390M  /iscsi/root
iscsi_rpool/rpool/ROOT/solaris@install 64K   -    390M  -
iscsi_rpool/rpool/ROOT/solaris/var 46.1M 9.36G 45.4M  /iscsi/
root/var
iscsi_rpool/rpool/ROOT/solaris/var@install 644K   - 45.4M  -
iscsi_rpool/rpool/VARSHARE 31K  9.36G   31K  /var/share
iscsi_rpool/rpool/export 62K  9.36G   31K  /export
iscsi_rpool/rpool/export/home 31K  9.36G   31K  /export/home

```

The new zone hosted on iSCSI-based shared storage resources has been successfully installed and can now be booted using `zoneadm boot`.

After the zone has been booted, the zone administrator observes virtualized ZFS datasets and storage pools from within the zone.

```

root@iscsi:~# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
data     9.94G   85K  9.94G   0%  1.00x  ONLINE  -
rpool    9.94G  449M  9.50G   4%  1.00x  ONLINE  -

root@iscsi:~# zpool status -v
pool: data
state: ONLINE
scan: none requested
config:

    NAME                                STATE      READ WRITE CKSUM
    data                                ONLINE      0   0   0
      mirror-0
        c0t600144F035FF850000050C884E80002d0  ONLINE      0   0   0
        c0t600144F035FF850000050C884EC0003d0  ONLINE      0   0   0

pool: rpool
state: ONLINE
scan: none requested
config:

    NAME                                STATE      READ WRITE CKSUM
    rpool                                ONLINE      0   0   0
      c0t600144F035FF850000050C884E50001d0  ONLINE      0   0   0

root@iscsi:~# zfs list -t all
NAME                                USED  AVAIL  REFER  MOUNTPOINT
data                                85K  9.78G   31K  /data
rpool                                464M  9.33G   31K  /rpool
rpool/ROOT                          464M  9.33G   31K  legacy
rpool/ROOT/solaris                   464M  9.33G  416M  /
rpool/ROOT/solaris@install           1.83M   -    390M  -
rpool/ROOT/solaris/var               46.2M  9.33G  45.6M  /var
rpool/ROOT/solaris/var@install       674K   -   45.4M  -
rpool/VARSHARE                       39K  9.33G   39K  /var/share
rpool/export                         96.5K  9.33G   32K  /export
rpool/export/home                    64.5K  9.33G   32K  /export/home
rpool/export/home/user               32.5K  9.33G   32.5K  /export/home/user

```

Example 13-7 Using DAS Storage Devices With Oracle Solaris Zones

This exercise uses direct attached local storage devices to configure and install a zone on Oracle Solaris. Note that this method is usually not portable across different hosts.

First, discover the available local disks with the `format` command. Then, use `suriadm lookup-uri` to construct the corresponding storage URIs to be used within the zone configuration.

```
root@host:~# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  1. c4t1d0 <SEAGATE-ST336704LSUN36G-0326-33.92GB>
     /pci@0,0/pci1022,7450@a/pci17c2,20@4/sd@1,0
  2. c4t2d0 <FUJITSU-MAT3073NC-0104-68.49GB>
     /pci@0,0/pci1022,7450@a/pci17c2,20@4/sd@2,0
  3. c4t3d0 <SEAGATE-ST336704LSUN36G-0326-33.92GB>
     /pci@0,0/pci1022,7450@a/pci17c2,20@4/sd@3,0
  4. c4t4d0 <FUJITSU-MAW3073NC-0103-68.49GB>
     /pci@0,0/pci1022,7450@a/pci17c2,20@4/sd@4,0

root@host:~# suriadm lookup-uri -t dev /dev/dsk/c4t1d0
dev:dsk/c4t1d0
root@host:~# suriadm lookup-uri -t dev /dev/dsk/c4t2d0
dev:dsk/c4t2d0
root@host:~# suriadm lookup-uri -t dev /dev/dsk/c4t3d0
dev:dsk/c4t3d0
root@host:~# suriadm lookup-uri -t dev /dev/dsk/c4t4d0
dev:dsk/c4t4d0
```

Using those storage URIs, configure a zone with a `rootzpool` and a `zpool` resource, both representing mirrored ZFS storage pools.

```
root@host:~# zonecfg -z disk
Use 'create' to begin configuring a new zone.
zonecfg:disk> create
create: Using system default template 'SYSdefault'
zonecfg:disk> add rootzpool
zonecfg:disk:rootzpool> add storage dev:dsk/c4t1d0
zonecfg:disk:rootzpool> add storage dev:dsk/c4t3d0
zonecfg:disk:rootzpool> end
zonecfg:disk> add zpool
zonecfg:disk:zpool> set name=dpool
zonecfg:disk:zpool> add storage dev:dsk/c4t2d0
zonecfg:disk:zpool> add storage dev:dsk/c4t4d0
zonecfg:disk:zpool> end
zonecfg:disk> commit
zonecfg:disk> exit
```

Now install the zone.

```
root@host:~# zoneadm -z disk install
Created zone zpool: disk_rpool
Created zone zpool: disk_dpool
Progress being logged to /var/log/zones/zoneadm.20130213T132236Z.disk.install
Image: Preparing at /disk/root.

AI Manifest: /tmp/manifest.xml.r0a0he
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
Zonename: disk
Installation: Starting ...
```

```

        Creating IPS image
Startup linked: 1/1 done
        Installing packages from:
            solaris
            origin: http://pkg.oracle.com/solaris/release/
DOWNLOAD          PKGS          FILES    XFER (MB)   SPEED
Completed         183/183    33556/33556  222.2/222.2  2.0M/s

PHASE              ITEMS
Installing new actions  46825/46825
Updating package state database      Done
Updating image state                 Done
Creating fast lookup database        Done
Installation: Succeeded
    
```

Note: Man pages can be obtained by installing pkg:/system/manual done.

Done: Installation completed in 308.358 seconds.

Next Steps: Boot the zone, then log into the zone console (zlogin -C) to complete the configuration process.

Log saved in non-global zone as /disk/root/var/log/zones/zoneadm.20130213T132236Z.disk.install

After zone installation, the following two new ZFS storage pools will be online.

```

root@host:/# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
disk_dpoo 68G   83.5K  68.0G  0%   1.00x  ONLINE  -
disk_rpool 33.8G  434M   33.3G  1%   1.00x  ONLINE  -
    
```

```

root@host:/# zpool status -v disk_rpool
pool: disk_rpool
state: ONLINE
scan: none requested
config:
    
```

NAME	STATE	READ	WRITE	CKSUM
disk_rpool	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c4t1d0	ONLINE	0	0	0
c4t3d0	ONLINE	0	0	0

```

root@host:/# zpool status -v disk_dpool
pool: disk_dpool
state: ONLINE
scan: none requested
config:
    
```

NAME	STATE	READ	WRITE	CKSUM
disk_dpool	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c4t2d0	ONLINE	0	0	0
c4t4d0	ONLINE	0	0	0

The zone installation will be entirely contained within this ZFS storage pool. The following ZFS dataset layout for this zone is present.

```
root@host:~# zfs list -t all|grep disk
disk_dpool                83.5K  66.9G   31K  /
disk_dpool
disk_rpool                434M  32.8G   32K  /disk
disk_rpool/rpool         433M  32.8G   31K  /rpool
disk_rpool/rpool/ROOT    433M  32.8G   31K  legacy
disk_rpool/rpool/ROOT/solaris 433M  32.8G  389M  /disk/
root
disk_rpool/rpool/ROOT/solaris@install 63K    -   389M  -
disk_rpool/rpool/ROOT/solaris/var    43.8M  32.8G  43.2M  /disk/
root/var
disk_rpool/rpool/ROOT/solaris/var@install 584K    -   43.2M  -
disk_rpool/rpool/VARSHARE           31K  32.8G   31K  /var/
share
disk_rpool/rpool/export           62K  32.8G   31K  /export
disk_rpool/rpool/export/home      31K  32.8G   31K  /
export/home
```

The new zone hosted on local device storage resources has been successfully installed and can now be booted using the `zoneadm boot` command.

After the zone has been booted, the zone administrator can observe virtualized ZFS datasets and storage pools from inside the zone.

```
root@disk:~# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
dpool    68G   83.5K  68.0G  0%   1.00x  ONLINE  -
rpool   33.8G   472M  33.3G  1%   1.00x  ONLINE  -
```

```
root@disk:~# zpool status -v
pool: dpool
state: ONLINE
scan: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
dpool	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c4t2d0	ONLINE	0	0	0
c4t4d0	ONLINE	0	0	0

```
pool: rpool
state: ONLINE
scan: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c4t1d0	ONLINE	0	0	0
c4t3d0	ONLINE	0	0	0

```
root@disk:~# zfs list -t all
NAME                                USED  AVAIL  REFER  MOUNTPOINT
dpool                                83.5K 66.9G   31K  /dpool
rpool                                465M  32.8G   31K  /rpool
rpool/ROOT                           465M  32.8G   31K  legacy
rpool/ROOT/solaris                   465M  32.8G  416M  /
```

```

rpool/ROOT/solaris@install      5.60M      -   389M      -
rpool/ROOT/solaris/var          43.9M     32.8G     43.3M     /var
rpool/ROOT/solaris/var@install   618K       -   43.2M      -
rpool/VARSHARE                  39K        32.8G      39K       /var/share
rpool/export                    96.5K      32.8G      32K       /export
rpool/export/home               64.5K      32.8G      32K       /export/home
rpool/export/home/user          32.5K      32.8G     32.5K     /export/home/user

```

Example 13-8 Using Fibre Channel-Based Storage Devices With Oracle Solaris Zones

This exercise uses a shared storage device provided over fibre channel to configure and install a zone on Oracle Solaris.

First, discover the fibre channel logical units currently visible to our host by using the `fcinfo lu` command.

```

root@host:~# fcinfo lu -v
OS Device Name: /dev/rdisk/c0t600144F0DBF8AF190000564C9AB9001Bd0s2
      HBA Port WWN: 21000024ff3ee788
      Controller: /dev/cfg/c2
            Remote Port WWN: 21000024ff3ee89e
            LUN: 0
            State: active/optimized
      Vendor: SUN
      Product: ZFS Storage 7120
      Device Type: Disk Device
      Unformatted capacity: 102400.000 MBytes

```

Use `suriadm lookup-uri` to construct a storage URI based on the device path. Remove the slice portion of the device name for the query to retrieve a storage URI representing an entire LU.

```

root@host:~# suriadm lookup-uri /dev/dsk/c0t600144F0DBF8AF190000564C9AB9001Bd0
dev:dsk/c0t600144F0DBF8AF190000564C9AB9001Bd0
lu:luname.naa.600144f0dbf8af190000564c9ab9001b
lu:initiator.naa.21000024ff3ee788,target.naa.21000024ff3ee89e,luname.naa.600144f0dbf8af
190000564c9ab9001b

```

From the three URIs displayed, select the `luname-only` form of the logical unit storage URI for use in the zone configuration.

```

root@host:~# zonecfg -z fc
Use 'create' to begin configuring a new zone.
zonecfg:fc> create
create: Using system default template 'SYSdefaultzonecfg:fc> add rootzpool
zonecfg:fc:rootzpool> add storage lu:luname.naa.600144f0dbf8af190000564c9ab9001b
zonecfg:fc:rootzpool> end
zonecfg:fc> commit
zonecfg:fc> exit

```

Install the zone.

```

root@host:~# zoneadm -z fc install
Created zone zpool: fc_rpool
Progress being logged to /var/log/zones/zoneadm.20130214T045957Z.fc.install
      Image: Preparing at /fc/root.

AI Manifest: /tmp/manifest.xml.K9aaow
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
      Zonename: fc
Installation: Starting ...

```

```

        Creating IPS image
Startup linked: 1/1 done
        Installing packages from:
            solaris
                origin:  http://pkg.oracle.com/solaris/release/
DOWNLOAD                PKGS            FILES      XFER (MB)   SPEED
Completed                190/190    34246/34246  231.3/231.3 7.2M/s

PHASE                    ITEMS
Installing new actions    48231/48231
Updating package state database      Done
Updating image state             Done
Creating fast lookup database        Done
Installation: Succeeded
    
```

Note: Man pages can be obtained by installing `pkg:/system/manual` done.

Done: Installation completed in 104.318 seconds.

Next Steps: Boot the zone, then log into the zone console (`zlogin -C`) to complete the configuration process.

Log saved in non-global zone as `/fc/root/var/log/zones/zoneadm.20130214T045957Z.fc.install`

After zone installation, the following new ZFS storage pool will be online.

```

root@host:~# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
fc_rpool  39.8G  441M   39.3G   1%  1.00x  ONLINE  -
    
```

```

root@host:~# zpool status -v fc_rpool
pool: fc_rpool
state: ONLINE
scan: none requested
config:
    
```

NAME	STATE	READ	WRITE	CKSUM
fc_rpool	ONLINE	0	0	0
c0t600144F0DBF8AF190000564C9AB9001Bd0	ONLINE	0	0	0

The zone installation will be entirely contained within this ZFS storage pool. The zone has the following ZFS dataset layout.

```

root@host:~# zfs list -t all|grep fc
fc_rpool                440M  38.7G   32K  /fc
fc_rpool/rpool          440M  38.7G   31K  /rpool
fc_rpool/rpool/ROOT     440M  38.7G   31K  legacy
fc_rpool/rpool/ROOT/solaris  440M  38.7G  405M  /fc/root
fc_rpool/rpool/ROOT/solaris@install  67K   -    405M  -
fc_rpool/rpool/ROOT/solaris/var  34.3M  38.7G  33.6M  /fc/root/var
fc_rpool/rpool/ROOT/solaris/var@install  665K   -    33.6M  -
fc_rpool/rpool/VARSHARE  31K   38.7G   31K  /var/share
fc_rpool/rpool/export   62K   38.7G   31K  /export
fc_rpool/rpool/export/home  31K   38.7G   31K  /export/home
    
```


The new zone hosted on shared storage provided from a fibre channel target has been successfully installed. This zone can now be booted using `zoneadm boot`.

After the zone has been booted, the zone administrator can observe virtualized ZFS datasets and storage pools from inside the zone.

```
root@fc:~# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool    39.8G   451M  39.3G   1%  1.00x  ONLINE  -

root@fc:~# zpool status -v
pool: rpool
state: ONLINE
scan: none requested
config:

          NAME                                     STATE      READ WRITE CKSUM
          rpool                                     ONLINE      0   0   0
          c0t600144F0DBF8AF190000564C9AB9001Bd0  ONLINE      0   0   0

root@fc:~# zfs list -t all
NAME                                     USED  AVAIL  REFER  MOUNTPOINT
rpool                                     467M  38.7G   31K    /rpool
rpool/ROOT                               467M  38.7G   31K    legacy
rpool/ROOT/solaris                       467M  38.7G  430M    /
rpool/ROOT/solaris@install                1.90M   -    405M    -
rpool/ROOT/solaris/var                   34.4M  38.7G  33.7M    /var
rpool/ROOT/solaris/var@install            703K   -    33.6M    -
rpool/VARSHARE                           39K   38.7G   39K    /var/share
rpool/export                             96.5K  38.7G   32K    /export
rpool/export/home                       64.5K  38.7G   32K    /export/home
rpool/export/home/user                   32.5K  38.7G   32.5K    /export/home/user
```

Migrating Oracle Solaris Zones Hosted on Shared Storage Resources

Migration of zones hosted on shared storage using storage URIs in the zone configuration is a straightforward and simple process. The only CLIs required for this process are the `zoneadm` and `zonecfg` commands. No additional commands must be run to migrate zones with storage resources or ZFS storage pools.

Example 13-9 Migrating a Zone Based on iSCSI Shared Storage

This example shows the steps to migrate the zone based on iSCSI shared storage from its current host to a new host.

```
root@initiator:/# zoneadm list -cp
0:global:running:/:solaris:shared:-:none
-:iscsi:installed:/iscsi:a0a4ba0d-9d6d-cf2c-cc42-f123a5e3ee11:solaris:excl:-:
```

The first step is to detach the zone on the current host. The zone will move from the installed state into the configured state. All the ZFS storage pools will be exported and the shared storage resources will be unconfigured automatically.

```
root@initiator:/# zoneadm -z iscsi detach
Exported zone zpool: iscsi_rpool
Unconfigured zone storage resource(s) from:
    iscsi://target/luname.naa.600144F035FF850000050C884E50001
```

```
Exported zone zpool: iscsi_data
Unconfigured zone storage resource(s) from:
    iscsi://target/lunname.naa.600144F035FF8500000050C884E80002
    iscsi://target/lunname.naa.600144F035FF8500000050C884EC0003
```

```
root@initiator:/# zoneadm list -cp
0:global:running://::solaris:shared:-:none
-:iscsi:configured:/iscsi::solaris:excl:-:
```

The last step required on the current host is to export the zone configuration with `zonecfg export`, and transfer the resulting file onto the new host.

```
root@initiator:/# zonecfg -z iscsi export -f /export/iscsi.cfg
```

On the new host, instantiate the zone configuration first from the saved file by using the `zonecfg` command. The zone will be in the `configured` state.

```
root@newhost:/# zonecfg -z iscsi -f /export/iscsi.cfg
```

```
root@newhost:/# zoneadm list -cp
0:global:running://::solaris:shared:-:none
-:iscsi:configured:/iscsi::solaris:excl:-:
```

Then, attach the zone on the new host with `zoneadm attach`. The zone will move into the `installed` state. All shared storage resources will be configured and the corresponding ZFS storage pools will be imported automatically.

```
root@newhost:/# zoneadm -z iscsi attach
Configured zone storage resource(s) from:
    iscsi://target/lunname.naa.600144F035FF8500000050C884E50001
Imported zone zpool: iscsi_rpool
Configured zone storage resource(s) from:
    iscsi://target/lunname.naa.600144F035FF8500000050C884E80002
    iscsi://target/lunname.naa.600144F035FF8500000050C884EC0003
Imported zone zpool: iscsi_data
Progress being logged to /var/log/zones/zoneadm.20130214T145001Z.iscsi.attach
Installing: Using existing zone boot environment
    Zone BE root dataset: iscsi_rpool/rpool/ROOT/solaris
    Cache: Using /var/pkg/publisher.
Updating non-global zone: Linking to image /.
Processing linked: 1/1 done
Updating non-global zone: Auditing packages.
No updates necessary for this image.

Updating non-global zone: Zone updated.
    Result: Attach Succeeded.
Log saved in non-global zone as /iscsi/root/var/log/zones/
zoneadm.20130214T145001Z.iscsi.attach
```

```
root@newhost:/# zoneadm list -cp
0:global:running://::solaris:shared:-:none
-:iscsi:installed:/iscsi:a19fbb45-4af3-670f-c58e-ee48757c75d6:solaris:excl:-:
```

The zone has been migrated to the new host and is now ready to be booted with `zoneadm boot`.

This process is basically the same for the three different types of storage URIs supported in Oracle Solaris. For iSCSI-based or fibre channel-based storage resources, you must also confirm that the new host has access to the same logical unit and target ports.

Moving Existing Zones To or From Shared Storage Zone Configurations

You can move an existing file system-based zone installation to a shared storage-based zone configuration. You can also move an installed zone from shared storage resources to a file system configuration.



Note:

Because this process involves moving a zone installation across storages, these actions are possible only on zones in the `installed` state.

In Oracle Solaris 11.4, the `zonepath` for zones that contain a `rootzpool` resource cannot be changed. See the [solaris\(7\)](#) man page for the specific defaults that the `solaris` brand supports.

The following procedures describe how to make these zone configuration changes:

- [How to Move a Zone to a Shared Storage Configuration](#)
- [How to Move a Zone From a Shared Storage Configuration](#)
- [How to Move a Zone to Different Shared Storage](#)
- [How to Add or Move a ZFS Storage Pool to an Installed Zone](#)

To move a zone that is not going to or from shared storage, go to [How to Move a Zone That Is Not on Shared Storage](#).

How to Move a Zone to a Shared Storage Configuration

Perform this procedure to move a `solaris` zone from a local file system to shared storage. The `zoneadm move` command provides options to either create a new ZFS storage pool or import an existing one for the zone to use.

If you migrate a non-ZOSS zone into a ZOSS configuration that uses a non-default `zonepath` property value, the property value resets to the system-derived default value from the `zonepath` template.

Note that you cannot specify a new `zonepath` property value.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Verify that the zone is in the installed state.

```
global# pfbash zoneadm list -cv
  ID NAME           STATUS    PATH                               BRAND  IP
  0 global          running   /                                 solaris shared
  - zonename
  installed /system/zones/zonename solaris excl
```

3. If the zone you want to move is in the running state, halt the zone.

After the zone is halted, verify that it is now in the `installed` state.

```
global$ zoneadm -z zonename halt
global# zoneadm list -cv
```

4. Reconfigure the zone to move it to shared storage.

```
global$ zoneadm -z zonename move -p URI [-p URI] [-x extended-option ...]
```

-p URI

Specifies a storage URI. Use a separate -p option for each additional URI to specify. For more information about storage URIs, see the [suri\(7\)](#) man page.

-x {force-zpool-import | force-zpool-create=rpool | create-size}

Specifies optionally creating or importing the ZFS storage pool that the zone will use. The -x option accepts only one argument per operation.

force-zpool-import

Imports an existing ZFS storage pool.

force-zpool-create=rpool

Specifies to create a new ZFS storage pool with the user-supplied name *rpool*.

create-size=size

Specifies to create new storage URI objects of the specified *size*. By default the value of *size* is in bytes, unless the value includes a format specifier such as 1M, 1G, and 1T.

The reconfigured zone remains in the `installed` state.

How to Move a Zone From a Shared Storage Configuration

Perform this procedure to move a zone that uses shared storage (ZOSS) to a configuration that uses a local file system. The `zoneadm move` command creates new target ZFS datasets that correspond to the specified new `zonename`. You can optionally unconfigure or remove the ZFS storage pool that the zone no longer uses.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Verify that the zone is in the installed state.

```
global$ pfbash zoneadm list -cv
  ID NAME           STATUS    PATH                               BRAND  IP
   0 global          running   /                                   solaris shared
  - zonename
installed /system/zones/zonename  solaris excl
```

3. If the zone you want to move is in the running state, halt the zone.

After the zone is halted, verify that it is now in the `installed` state.

```
global$ pfbash zoneadm -z zonename halt
global$ zoneadm list -cv
```

4. Reconfigure the zone to move it to a local file system.

```
global$ zoneadm -z zonename move -x remove-rootzpool [-x extended-option ...]
```

```
-x {remove-rootzpool | force-zpool-destroy=rpool | force-storage-destroy-
all}
```

Specifies extended options to process the `rootzpool` resource of the zone after moving the zone out of shared storage. The `-x` option can take only one extended option per operation, but you can specify more than one `-x` option in a single command.

remove-rootzpool

Required to move a zone from shared storage. Specifies removing the `rootzpool` resource from the zone configuration.

force-zpool-destroy=rpool

Specifies exporting and then destroying the ZFS storage pool that is specified by the `rootzpool` resource.

force-storage-destroy-all

Specifies destroying the storage objects in the ZFS storage pool that is specified by the `rootzpool` resource.

Note that you cannot specify a new zone path to which to move the zone.

The reconfigured zone remains in the `installed` state.

How to Move a Zone to Different Shared Storage

Ensure that the destination zone to which you move the shared storage already exists and contains a `zpool`.

Perform this procedure to move a zone to different shared storage. The zone configuration remains otherwise the same.

1. Become a zone administrator.

For more information, see [Using Rights Profiles to Install and Manage Zones](#).

2. Verify that the zone is in the installed state.

```
global$ pfbash zoneadm list -cv
  ID NAME          STATUS  PATH          BRAND  IP
   0 global          running /             solaris shared
  - zonename
installed /system/zones/zonename solaris excl
```

3. If the zone you want to move is in the running state, halt the zone.

After the zone is halted, verify that it is now in the `installed` state.

```
global$ zoneadm -z zonename halt
global$ zoneadm list -cv
```

4. Change the location of the shared storage that the zone uses.

```
global$ zoneadm -z zonename move -p URI [-p URI ...] \
-x remove-rootzpool [-x extended-option ...] [new-zonepath]
```

-p URI

Specifies a storage URI. Use a separate `-p` option for each additional URI to specify. For more information about storage URIs, see the [suri\(7\)](#) man page.

```
-x {remove-rootzpool | force-zpool-destroy=rpool | force-storage-destroy-all}
```

Specifies extended options to process the `rootzpool` resource of the zone after moving the zone out of shared storage. The `-x` option can take only one extended option per operation, but you can specify more than `-x` option in a single command.

remove-rootzpool

Required to move a zone from shared storage. Specifies removing the `rootzpool` resource from the zone configuration.

force-zpool-destroy=rpool

Specifies exporting the ZFS storage pool that is specified by the `rootzpool` resource.

force-storage-destroy-all

Specifies destroying the storage objects in the ZFS storage pool that is specified by the `rootzpool` resource.

[**new-zonepath**]

Specifies an optional new zone path to which to move the zone.

How to Add or Move a ZFS Storage Pool to an Installed Zone

Ensure that the ZFS storage pool to add or move is created.

Ensure that the storage URI is available on the host system. For more information, see [Managing Storage URIs and Shared Storage Resources](#) and the `suriadm(8)` man page.

Perform this procedure to do either of the following tasks:

- Add an existing ZFS storage pool to an installed zone.
- Migrate a ZFS storage pool from one installed zone to another.

1. Assume the root role.

For more information, see [Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*](#).

2. Shut down the zone.

```
global$ pfbash zoneadm -z zonename shutdown
```

3. Detach the zone.

```
global$ zoneadm -z zonename detach
```

The zone moves to the `configured` state.

4. Add a new zpool resource and corresponding shared storage resources to the zone configuration.

```
global$ zonecfg -z zonename
zonecfg:zonename> add zpool
zonecfg:zonename:zpool> add storage URI-string
zonecfg:zonename:zpool> set name=name-string
zonecfg:zonename:zpool> end
```

5. Reattach the zone.

```
global$ zoneadm -z zonename attach
```

The Zones framework configures the shared storage resources, imports the existing ZFS storage pool, and assigns it to the installed zone. The zone moves to the `installed` state.

6. Boot the zone.

```
global$ zoneadm -z zonename boot
```

The zone moves to the `running` state.

Zones Documentation References for Shared Storage Resources

For more information about zones on shared storage, refer to the following sources of information.

- [beadm\(8\)](#) man page
- [fcinfo\(8\)](#) man page
- [iscsiadm\(8\)](#) man page
- [iser\(4D\)](#) man page
- [itadm\(8\)](#) man page
- [sasinfo\(8\)](#) man page
- [stmfadm\(8\)](#) man page
- [suri\(7\)](#) man page
- [suriadm\(8\)](#) man page
- [solaris\(7\)](#) man page
- [zones\(7\)](#) man page
- [zoneadm\(8\)](#) man page
- [zonecfg\(8\)](#) man page
- [zpool\(8\)](#) man page
- [Oracle Solaris Zones Configuration Resources](#)

Glossary

brand

An instance of the BrandZ functionality, which provides non-global zones that contain non-native operating environments used for running applications.

branded zone

An isolated environment in which to run non-native applications in non-global zones.

cap

A limit that is placed on system resource usage.

capping

The process of placing a limit on system resource usage.

datalink

An interface at Layer 2 of the OSI protocol stack, which is represented in a system as a STREAMS DLPI (v2) interface. This interface can be plumbed under protocol stacks such as TCP/IP. In the context of Oracle Solaris 10 zones, data-links are physical interfaces, aggregations, or VLAN-tagged interfaces. A datalink can also be referred to as a physical interface, for example, when referring to a NIC or a VNIC.

default pool

The pool created by the system when pools are enabled.

See also [resource pool](#).

default processor set

The processor set created by the system when pools are enabled.

See also [processor set](#).

dynamic configuration

Information about the disposition of resources within the resource pools framework for a given system at a point in time.

dynamic reconfiguration

On SPARC based systems, the ability to reconfigure hardware while the system is running. Also known as DR.

extended accounting

A flexible way to record resource consumption on a task basis or process basis in the Oracle Solaris operating system.

fair share scheduler

A scheduling class, also known as FSS, that allows you to allocate CPU time that is based on shares. Shares define the portion of the system's CPU resources allocated to a project.

FSS

See [fair share scheduler](#).

global administrator

The root user or an administrator with the `root` role. When logged in to the global zone, the global administrator or a user granted the appropriate authorizations can monitor and control the system as a whole.

See also [zone administrator](#).

global scope

Actions that apply to resource control values for every resource control on the system.

global zone

The zone contained on every Oracle Solaris system. When non-global zones are in use, the global zone is both the default zone for the system and the zone used for system-wide administrative control.

See also [non-global zone](#).

immutable zone

A zone configured with a read-only root.

local scope

Local actions taken on a process that attempts to exceed the control value.

locked memory

Memory that cannot be paged.

memory cap enforcement threshold

The percentage of physical memory utilization on the system that will trigger cap enforcement by the resource capping daemon.

naming service database

The name service information database for both LDAP containers and local files.

native zones

`solaris` branded non-global zones. Zones of other brands, such as the `labeled` and `solaris10`, or `solaris-kz` brands are not native zones.

The operating system level within a running native zone always matches the level of the global zone. Because the same kernel is used, no system-call translation is performed.

Note that native zones are branded `solaris` in Oracle Solaris 11 and `native` in Oracle Solaris 10.

non-global zone

A virtualized operating system environment created within a single instance of the Oracle Solaris operating system. The Oracle Solaris Zones software partitioning technology is used to virtualize operating system services. The packages in a non-global zone are identical to the global zone packages because the zones share the same kernel.

non-global zone administrator

See [zone administrator](#).

Oracle Solaris Zones

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

pool

See [resource pool](#).

pool daemon

The `poold` system daemon that is active when dynamic resource allocation is required.

project

A network-wide administrative identifier for related work.

processor set

A disjoint grouping of CPUs. Each processor set can contain zero or more processors. A processor set is represented in the resource pools configuration as a resource element. Configured using the `pset` resource pool element.

resident set size

The size of the resident set. The resident set is the set of pages that are resident in physical memory.

resource

An aspect of the computing system that can be manipulated with the intent to change application behavior.

resource capping daemon

A daemon that regulates the consumption of physical memory by processes running in projects that have resource caps defined.

resource consumer

Fundamentally, an Oracle Solaris process. Process model entities such as the project and the task provide ways of discussing resource consumption in terms of aggregated resource consumption.

resource control

A per-process, per-task, or per-project limit on the consumption of a resource.

resource management

A functionality that enables you to control how applications use available system resources.

resource partition

An exclusive subset of a resource. All of the partitions of a resource sum to represent the total amount of the resource available in a single executing Oracle Solaris instance.

resource pool

A configuration mechanism that is used to partition machine resources. A resource pool represents an association between groups of resources that can be partitioned.

resource set

A process-bindable resource. Most often used to refer to the objects constructed by a kernel subsystem offering some form of partitioning. Examples of resource sets include scheduling classes and processor sets.

RSS

See [resident set size](#).

scanner

A kernel thread that identifies infrequently used pages. During low memory conditions, the scanner reclaims pages that have not been recently used.

static pools configuration

A representation of the way in which an administrator would like a system to be configured with respect to resource pools functionality.

task

In resource management, a process collective that represents a set of work over time. Each task is associated with one project.

working set size

The size of the working set. The working set is the set of pages that the project workload actively uses during its processing cycle.

workload

An aggregation of all processes of an application or group of applications.

ZBE

Zone boot environment.

zone administrator

The rights of a zone administrator are confined to a non-global zone.

See also [global administrator](#).

zone state

The status of a non-global zone. The zone state is one of configured, incomplete, installed, ready, unavailable, running, or shutting down.

Index

Symbols

/etc/vfstab file, [7-10](#)

A

add dataset configuration

example, [3-11](#)

exception to MWAC, [11-3](#)

live mode, [3-13](#)

temporarily, [3-13](#)

add fs configuration

example, [3-4](#)

exception to MWAC, [11-3](#)

UFS example, [3-4](#)

adding

dataset, [3-11](#), [3-13](#)

file system, [3-4](#)

privileges to `limitpriv`, [12-5](#)

system daemons to TPD, [11-6](#)

ZFS storage pool, [13-26](#)

administering

device drivers, [7-4](#)

immutable zones, [11-5](#)

packages, [6-1](#)

shared storage resources, [13-3](#)

writable immutable zones, [11-8](#)

zones delegated restarter, [5-5](#)

alternate root environment, [6-7](#)

applications

devices and zones, [7-3](#)

privileges and zones, [7-2](#)

unsuitable for zones, [7-2](#), [7-3](#)

zone backups, [7-24](#)

archiveadm command

creating a recovery archive, [8-5](#)

example, [8-5](#)

installing zone from, [2-7](#)

zone transformation and, [10-2](#)

zone transformations and, [7-19](#)

assigning

boot priority, [5-4](#)

CPUs to zones, [3-3](#)

CPUs to zones from global zone, [3-13](#)

devices to zones, [7-3](#)

assigning (*continued*)

FSS shares, [7-5](#)

rights profiles to zone administrators, [1-5](#)

rights to migrate zones, [8-1](#)

shared storage to zones, [13-5](#)

attaching zones

shared storage options, [13-8](#)

autoboot property, [2-1](#), [3-7](#), [8-5](#)

AutoFS file system

security restrictions, [7-11](#)

B

backing up

databases in zones, [7-25](#)

LOFS directories, [7-23](#)

zone contents, [7-24](#)

zones, [7-23](#), [7-26](#)

beadm command

boot environments and shared storage,
[13-10](#)

in non-global zones, [6-5](#)

boot environments

alternate root, [6-7](#)

effect of shared zone state, [6-7](#)

orphaned, [6-7](#)

zones and, [6-5](#)

boot order

assigning priority, [5-4](#)

booting

immutable zone, [11-8](#)

zones in order, [5-4](#)

branded zones, [1-2](#)

brands

list of, [1-2](#)

C

canceling

zone configuration, [3-7](#)

cap limits for memory, [3-4](#)

changing

configuration temporarily, [3-7](#)

property values, [3-3](#)

resource control settings, [3-7](#)

changing (*continued*)
 resources, [3-7](#)
 shared storage URI, [13-7](#)
 shared-IP to exclusive-IP, [4-5](#)
 zone properties, [3-7](#)

cloning zones
 for small number of zones, [2-7](#)
 how to, [2-8](#)
 shared storage options, [13-8](#)

cold migration
 overview, [7-17](#)
 zones, [8-3](#)

command references
 zones, [7-26](#)
 zones on shared storage, [13-27](#)

configuring
 immutable global zones, [11-9](#)
 modifying properties temporarily, [3-7](#)
 proxies to package repository, [6-2](#)
 zones
 default mode, [3-1](#)
 how to, [2-1](#)
 live mode, [3-1](#), [3-7](#)
 properties, [3-1](#)
 zpool on shared storage, [13-10](#)

converting
 global zone to `solaris` zone, [10-1](#)

core files and zones, [7-8](#)

CPUs
 assigning to zones, [3-3](#)
 assigning to zones from global zone, [3-13](#)
 monitoring, [7-7](#)
 reporting, [5-7](#)
 reporting utilization, [5-7](#)
`zone.cpu-shares` property, [7-5](#)

creating
 recovery archive, [8-5](#)
 zone administrator, [8-1](#)
 zone migration administrator, [8-1](#)
 zones, [3-1](#)

D

daemons
 adding to TPD, [11-6](#)
`zoneadmd`, [6-5](#)
`zonestatd`, [7-7](#)

datalinks
 exclusive-IP zones, [7-14](#)
 zones and, [4-5](#)

datasets
 adding to a running zone, [9-7](#)
 Live Reconfiguration, [9-7](#)
 removing from zone, [3-11](#), [9-9](#)

delegated restarter, [5-5](#)

deleting
 file systems, [2-10](#)

`deny-zbe-clone` option, [6-6](#)

description property, [2-1](#)

`destroy-orphan-zbes` option, [6-6](#)

destroying
 file systems, [2-10](#)

`devfs` file system, [7-3](#)

device drivers
 administering, [7-4](#)

devices and zones, [7-3](#)

directories
 sharing with global zone, [3-4](#)

disk space requirements, [1-1](#)

displaying
 list of zones, [2-5](#)
 name of current zone, [2-4](#)
 zone configuration information, [2-4](#)
 zone information from the global zone, [2-4](#),
 [2-6](#)
 zone processes, [7-6](#)
 zone resource properties, [2-5](#)
 zones delegated restarter, [5-5](#)

`dladm reset-linkprop` error message, [12-2](#)

DTrace Toolkit rights profile, [12-5](#)

`dtrace_proc` privilege, [7-8](#), [12-5](#)

`dtrace_user` privilege, [7-8](#), [12-5](#)

`dynamic-zones` MWAC security policy, [11-2](#)

E

examples
 adding DTrace rights to a zone, [12-5](#)
 authorizing user to configure zones, [8-1](#)
 authorizing user to migrate all zones, [8-1](#)
 backing up zone with unified archives, [7-26](#)
 changing resource control, [3-7](#)
 changing `zone.cpu-shares` dynamically, [5-10](#)
 configuring private network, [4-1](#)
 creating a zone, [2-1](#)
 creating a zone with default values, [2-1](#)
 creating zones from a source zone, [2-8](#)
 discarding configuration changes, [3-7](#)
 dry run of cold migration, [8-3](#)
 listing default processor set, [5-7](#)
 making the global zone immutable, [11-2](#)
 migrating using iSCSI shared storage, [13-21](#)
 producing storage URIs, [13-3](#)
 proxy configuration in global zone, [6-2](#)
 reverting zone to default values, [3-7](#)
 showing CPU use, [5-7](#)
 showing per-zone filesystem activity, [5-9](#)
 showing zone network bandwidth from global
 zone, [5-7](#)

- examples (*continued*)
 - using DAS storage devices, [13-11](#)
 - using fibre channel-based storage devices, [13-11](#)
 - using iSCSI-based shared storage, [13-11](#)
 - using live mode, [3-7](#)
 - zones on shared storage, [13-11](#)
- exclusive-IP zones
 - network address, [1-2](#)
 - network bandwidth, [5-7](#)
 - reporting bandwidth utilization, [5-7](#)
- exiting a zone, [2-1](#)
- exporting
 - dataset prior to migration, [3-11](#)
- extended accounting and zones, [7-6](#)

F

- failsafe mode login, [12-6](#)
- fair share scheduler (FSS)
 - assigning shares, [7-5](#)
 - zones and, [5-10](#), [7-5](#)
- file systems
 - access restrictions from global zone, [7-13](#)
 - adding, [3-4](#)
 - adding UFS, [3-4](#)
 - administering `root` file system, [6-1](#)
 - archiving, [10-2](#)
 - destroying, [2-10](#)
 - `devfs`, [7-3](#)
 - `fs` resource, [3-4](#)
 - LOFS, [3-4](#)
 - mounting behavior, [7-11](#)
 - mounting options, [7-9](#)
 - mounting read-only, [11-3](#)
 - read-only `root` file system, [11-1](#)
 - report example, [5-9](#)
 - reporting activity, [5-9](#)
 - `root` directory, [7-9](#)
 - `root` zone path, [2-1](#)
 - security restrictions, [7-11](#)
 - temporarily writable, [11-8](#)
 - traversing, [7-11](#)
 - unmount failure, [12-5](#)
 - unmounting in zones, [7-10](#)
 - zones and, [7-9](#)
- `file-mac-profile` security property, [11-2](#)
- firewall
 - exclusive-IP zones, [7-15](#)
 - shared-IP zones, [7-16](#)
- fixed-configuration MWAC security policy, [11-2](#)
- flexible-configuration MWAC security policy, [11-2](#)

- `force-storage-destroy-all` option, [13-10](#)
- `force-zbe-clone` option, [6-7](#)
- `force-zpool-create-all` option, [13-9](#)
- `force-zpool-create=` option, [13-9](#)
- `force-zpool-destroy-all` option, [13-10](#)
- `force-zpool-destroy=zpoolname` option, [13-9](#)
- `force-zpool-import` option, [13-9](#)
- `fs` resource, [3-4](#)
- `fs-allowed` property, [7-9](#)
- FSS, [7-1](#)
- `fsstat` statistics, [5-9](#), [7-8](#)
- `fstype` statistics and zones, [5-9](#)

G

- global zones
 - caution about zone configuration from, [3-13](#)
 - configuring non-global zones from, [3-13](#)
 - device utilities not allowed in zones, [7-4](#)
 - directories shared with non-global zone, [3-4](#)
 - displaying zone information, [2-4](#)
 - `hwprovider` property, [3-6](#)
 - making immutable, [11-1](#)
 - monitoring zones, [7-6](#)
 - package differences from zones, [6-1](#)
 - privileges not allowed to zones, [7-2](#)
 - proxies to access package repositories, [6-2](#)
 - security measures and zones, [7-1](#)
 - time value differences in non-global zones, [3-6](#)
 - transforming into `solaris` zone, [10-1](#)
- goal services
 - Service Management Facility (SMF), [5-4](#)

H

- halting a zone, [12-5](#)
 - troubleshooting, [12-5](#)
- host names for zones, [1-2](#)
- `hwprovider` global property, [3-6](#)

I

- immutable zones
 - about, [11-1](#)
 - access with `zlogin` command, [11-8](#)
 - add dataset configuration, [11-3](#)
 - add `fs` configuration, [11-3](#)
 - administering, [11-5](#), [11-6](#)
 - authorization for `zlogin` command, [11-8](#)
 - booting writable, [11-8](#)
 - configuring, [11-1](#)
 - dynamic-zones, [11-2](#)

immutable zones (continued)

- file-mac-profile security property, [11-2](#)
- fixed-configuration MWAC policy, [11-2](#)
- flexible-configuration MWAC policy, [11-2](#)
- global zones, [11-1](#), [11-8](#)
- kernel zones, [11-1](#)
- non-global zones, [11-1](#)
- none MWAC policy, [11-2](#)
- setting security policy, [11-2](#)
- SMF commands and, [11-3](#)
- strict MWAC policy, [11-2](#)
- Trusted Path Domain, [11-6](#)

installadm command

- zone transformations and, [7-19](#)

installing zones

- clones and, [2-7](#)
- from archive, [2-7](#)
- overview, [2-1](#)
- shared storage options, [13-8](#)
- troubleshooting, [12-1](#)
- troubleshooting packages, [12-1](#)
- Unified Archives and, [2-7](#)

IPMP

- exclusive-IP zones, [7-15](#)
- shared-IP zones, [7-16](#)
- zones and, [4-3](#)

IPsec

- exclusive-IP zones, [7-15](#)
- shared-IP zones, [7-16](#)

iSCSI shared storage

- migrating zones, [13-21](#)

K**kernel zones**

- making immutable, [11-1](#)
- solaris-kz brand, [1-2](#)

L**labeled brand, [1-2](#)****limiting**

- administrator rights in zones, [1-5](#), [8-1](#)
- memory available to zones, [3-4](#)
- resources provided to zones, [3-3](#)
- zone size, [1-1](#)

limitpriv privilege set

- adding to, [12-5](#)
- DTrace privileges, [7-8](#)
- insufficient privileges, [12-5](#)
- troubleshooting, [12-5](#)

live migration

- by non-root users, [8-1](#)

Live Reconfiguration

- datasets, [9-7](#)

live zone editing fails, [12-2](#)**Live Zone Reconfiguration**

- adding a dataset, [9-7](#)
- dry run, [9-2](#)
- non-global zones, [9-1](#)
- procedures, [9-3](#)
- reloading the configuration, [9-3](#)
- temporary changes, [9-2](#)
- test run, [9-2](#)

LOFS file system

- backing up directories, [7-23](#)
- security restrictions, [7-11](#)
- sharing directory with global zone, [3-4](#)

logging in

- failsafe mode, [12-6](#)
- zone console, [2-1](#)

logging out of a zone, [2-1](#)**M****maintaining**

- immutable global zones, [11-9](#)

managing, [1-1](#)**mandatory write access control, [11-1](#)****memory**

- limiting use by zones, [3-4](#)
- reporting utilization, [5-7](#)
- zone requirements, [1-1](#)

migrate command, [7-17](#)**migrating**

- by non-root users, [8-1](#)
- iSCSI shared storage, [13-21](#)
- solaris zone, [8-3](#)
- transformation requirements, [7-20](#)
- zone without shared storage, [7-19](#)
- zones, [8-1](#)
- zones on shared storage, [13-21](#)

mknod command and zones, [7-13](#)**MNTFS file system**

- security restrictions, [7-11](#)

modifying, [1-1](#)**monitoring**

- active zone statistics, [7-7](#)
- resource use, [7-6](#)
- zone statistics, [7-8](#)
- zones from global zone, [7-6](#)

mounting file systems, [7-9](#)**moving a ZFS storage pool, [13-26](#)****moving a zone**

- from shared storage, [13-24](#)
- on shared storage, [13-23](#)
- on the same system, [8-6](#)
- to different shared storage, [13-25](#)

moving a zone (*continued*)
to shared storage, [13-23](#)

MWAC

definition, [11-1](#)
security policy exception, [11-3](#)
security policy for zones, [11-2](#)

N

namespaces

devices, for, [7-3](#)
zones and global zones, [7-11](#)

network address

exclusive-IP zone, [1-2](#)

network bandwidth

exclusive-IP zones, [5-7](#)
zones statistics, [5-7](#)

network interfaces

shared-IP zones, [7-16](#)

networking

exclusive-IP zones, [7-14](#)
shared-IP, [7-15](#)

new features

hwprovider global property, [3-6](#)

NFS file system

security restrictions, [7-11](#)
server, [7-9](#)

node names and zones, [7-8](#)

non-global zones

differences from global zone
available privileges, [7-2](#)
devices available, [7-3](#)
IKE in shared-IP zones, [7-16](#)
packages, [6-1](#)
privileges, [7-2](#)

none MWAC security policy, [11-2](#)

O

orphaned boot environments, [6-7](#)

overriding global zone proxies, [6-2](#)

P

packages

administering, [6-1](#)
system/file-system/ufs, [7-9](#)
troubleshooting zone installation, [12-1](#)
zones and, [6-1](#)

parallel zone updates, [6-3](#)

partitioning

exclusive-IP zones, [7-14](#)
shared-IP zones, [7-15](#)

persistent Live Zone Reconfiguration
applying to live configuration, [9-2](#)

pfbash shell, [1-5](#)

pfexec command, [1-5](#), [5-9](#)

physical property, [2-1](#)

populating a zone, [6-3](#)

privileges

missing in zone, [12-5](#)
troubleshooting, [12-5](#)
zone subset, [7-2](#)

processes

global visible to zone, [7-6](#)
visibility in zones, [7-7](#)

processor set

reporting information, [5-7](#)

PROCFS file system

security restrictions, [7-11](#)

properties

setting values, [3-3](#)

proxy

configuring for package repository, [6-2](#)
setting in global zone, [6-2](#)

pset resource pool element

reporting information, [5-7](#)

R

read-only zone root, [11-1](#)

recovery archive

creating, [8-5](#)
creating per zone, [7-20](#)
failing to install, [10-2](#)

recreating zones, [7-26](#)

reloading live zone configurations, [9-3](#)

removing

dataset resource, [3-11](#), [9-9](#)

renaming

zones on shared storage, [13-7](#)

reporting

file system activity, [5-9](#)

resource controls

changing settings, [3-7](#)
zones and, [7-5](#)

resources

exceptions to MWAC, [11-3](#)
modifying, [3-7](#)
modifying settings, [3-7](#)
reporting use of, [5-7](#)
setting values, [3-3](#)

restoring zones, [7-25](#)

restricting, [3-1](#)

reverting to zone default values, [3-7](#)

rights profiles

DTrace Toolkit, [12-5](#)
zone administration, [1-5](#)

rights profiles (*continued*)
 Zone Migration, 8-1
 Zone Security, 11-2

root directory, 7-9

root file system
 administering from global zone, 6-1

root zone path, 2-1

rootzpool resource, 13-6

running DTrace in a zone, 7-8, 12-5

S

security issues
 assigning devices to zones, 7-3
 file systems, 7-11
 global zones and zones, 7-1
 mounting non-default file systems, 7-9

security policy
 MWAC, 11-1

Service Management Facility (SMF)
 boot order, 5-4
 dependencies, 5-1
 goal services, 5-4
 immutable zone and, 11-3
 instance service state, 5-2
 overview, 5-1
 service state descriptions, 5-3
 status of services in zones, 5-6
 system-repository service, 6-2
 system/zones:default service, 5-1
 zones delegated restarter, 5-5
 displaying, 5-5
 properties description, 5-6
 setting properties, 5-5

setting properties
 autoboot, 3-7
 canceling set values, 3-7
 hwprovider, 3-6
 memory limits, 3-4
 time values, 3-6
 values, 3-3
 zones delegated restarter, 5-5

shared storage
 assigning to zones, 13-5
 automated ZFS, 13-7
 hosting zones, 13-10
 managing, 13-3
 migrating zones, 13-21
 moving zones to or from, 13-23
 unavailable zone state and, 12-3, 13-8
 URIs, 13-1
 zoneadm options, 13-8

sharing
 directory with global zone, 3-4

shutting down a zone, 2-10

SMF, 5-1

snapshots
 ZFS, 2-8

solaris brand, 1-2

solaris branded zone
 migrating, 8-3
 packages and, 6-1

solaris-kz brand, 1-2

solaris10 brand, 1-2

states
 zone SMF instance service, 5-3

storage property, 13-5

storage-based file systems
 security restrictions, 7-11

strict MWAC security policy, 11-2

system memory and zones, 1-1

system-repository proxy, 6-2

system-repository SMF service, 6-2

system/zones:default SMF service
 dependencies, 5-1

T

temporary Live Zone Reconfiguration, 9-2

test run
 Live Zone Reconfiguration, 9-2

time values and zones, 3-6

TPD, 11-1

traffic between zones
 exclusive-IP zones, 7-15
 shared-IP zones, 7-16

transformations
 starting with archiveadm, 10-2
 using archives, 7-19
 using zonep2vchk, 7-21
 zones to other types of zones, 10-1

troubleshooting
 dladm reset-linkprop error message, 12-2
 installation failure, 12-1
 live mode failure, 12-2
 missing privileges in zone, 12-5
 package installation failure, 12-1
 unavailable zone state, 12-3
 zone access denied, 12-6
 zone will not halt, 12-5
 zones, 12-1

Trusted Path Domain
 immutable zone access, 11-6
 immutable zones and SMF services, 11-3

U

UFS non-default file system
 assigning to zone, [3-4](#)
 mounting, [7-9](#)

unavailable zone state
 shared storage and, [12-3](#), [13-8](#)

unbootable boot environments
 zones, [6-6](#)

Unified Archive
 overview, [2-7](#)

uninstalling
 shared storage options, [13-9](#)
 zones, [2-10](#)

Universal Resource Identifiers (URIs), [13-1](#)

UNIX file system, [3-1](#)

unmounting
 /etc/vfstab file and, [7-10](#)
 unmounting file systems, [7-10](#)

updating zones in parallel, [6-3](#)

URIs
 shared storage, [13-1](#)

UUIDs
 zones, for, [12-3](#)

V

virtual zpool
 export, [9-10](#)

virtual-cpu resource type
 removing VCPUs in live configuration, [9-4](#)

Z

ZFS file system
 adding dataset, [3-11](#)
 clones, [2-8](#)
 security restrictions, [7-11](#)
 snapshots, [2-8](#)

ZFS storage pools
 adding to a zone, [13-26](#)
 moving to another zone, [13-26](#)

zlogin command
 displaying information inside zones, [2-6](#)
 failsafe mode, [12-6](#)
 zone console, [2-1](#)

zone administrator rights, [1-5](#)

zone configuration
 displaying, [2-4](#)

zone console login, [2-1](#)

zone construction, [6-3](#)

zone host name, [1-2](#)

Zone Migration rights profile, [8-1](#)

zone name
 displaying, [2-4](#)
 exclusive-IP zone and, [1-2](#)
 extended accounting tag, [7-6](#)
 name services difference between global and
 zone, [7-7](#)
 node name and, [7-8](#)
 zone UUID and, [12-3](#)

zone network address, [1-2](#)

zone requirements
 disk space, [1-1](#)
 hardware, [1-1](#)
 host names, [1-2](#)
 memory, [1-1](#)
 networking considerations, [1-2](#)

zone root directory, [7-9](#)

Zone Security rights profile, [11-2](#)

zone utilization, reporting, [5-7](#)

zone.cpu-shares property, [5-10](#), [7-5](#)

zoneadm apply command, [3-1](#)

zoneadm command
 about, [1-3](#)
 halt failure, [12-5](#)
 installing default zone, [3-1](#)
 installing zones, [2-1](#)
 listing zones, [2-5](#)
 shared storage options, [13-8](#)
 specifying alternate root, [6-7](#)
 viewing changes in boot environment, [6-5](#)
 zone transformations and, [7-19](#)

zoneadm install command, [2-1](#)

zoneadm migrate command, [7-18](#), [8-3](#)

zoneadm shutdown command, [8-3](#)

zoneadm uninstall command, [2-10](#)

zoneadmd daemon, [6-5](#)

zonecfg command, [9-1](#)
 about, [1-3](#)
 configuring zones, [3-1](#)
 default mode, [9-1](#)
 displaying resource properties, [2-5](#)
 info subcommand, [2-5](#)
 live mode, [9-1](#)
 zone transformations and, [7-19](#)

zonename command, [2-4](#)

zonep2vchk command
 analysis of conversion impact, [10-1](#)
 runtime check, [10-1](#)
 template generation, [10-1](#)
 transformation preparation, [7-21](#)
 zone transformations and, [7-19](#)

zonepath property, [2-5](#)

zones, [1-1](#)
 active monitoring, [7-7](#)
 administering datalinks, [4-5](#)

zones (*continued*)

- administering immutable, [11-5](#)
- applications and devices, [7-3](#)
- applications and privileges, [7-2](#)
- backing up, [7-23](#)
- backing up applications, [7-24](#)
- backing up files, [7-26](#)
- backing up individually, [7-24](#)
- backing up zone contents, [7-24](#)
- beadm actions on unbootable environments, [6-6](#)
- beadm command and, [6-5](#)
- boot environments, [6-5](#)
- brands, [1-2](#)
- canceling configuration, [3-7](#)
- changing resource control setting, [3-7](#)
- changing shared-IP to exclusive-IP, [4-5](#)
- commands used in, [7-26](#)
- configuring global immutable, [11-9](#)
- core files, [7-8](#)
- database backups, [7-25](#)
- device use, [7-3](#)
- displaying name, [2-4](#)
- exporting, [3-11](#)
- extended accounting, [7-6](#)
- fair share scheduler (FSS), [5-10](#), [7-5](#)
- file systems, [7-9](#)
- fsstat monitoring, [5-9](#)
- fstype statistics, [5-9](#)
- general administration, [4-1](#), [5-1](#)
- immutable global, [11-8](#)
- immutable zones, [11-1](#)
- IPMP, [4-3](#)
- IPsec, [7-15](#)
- limiting resources, [3-3](#)
- limiting size, [1-1](#)
- live reconfiguration, [9-1](#)
- maintaining global immutable, [11-9](#)
- migrating, [7-20](#), [8-1](#)
- migrating from unusable system, [8-4](#)
- modifying properties, [3-7](#)
- monitoring, [7-8](#)
- network bandwidth statistics, [5-7](#)
- networking and exclusive-IP, [7-14](#)
- networking and shared-IP, [7-15](#)
- NFS server, [7-9](#)
- node name, [7-8](#)
- observability, [7-7](#)
- orphaned boot environments, [6-7](#)
- packaging, [6-1](#)
- privileges, [7-2](#)
- process ID visibility, [7-7](#)
- processes visible to global zone, [7-6](#)
- proxy configuration, [6-2](#)
- recreating individual, [7-26](#)

zones (*continued*)

- resource controls, [7-5](#)
- resource exceptions to MWAC, [11-3](#)
- restoring, [7-25](#)
- reverting property value to default, [3-7](#)
- running DTrace in, [7-8](#)
- setting file-mac-profile value, [11-2](#)
- setting MWAC security policy, [11-2](#)
- setting property values, [3-3](#)
- setting resource values, [3-3](#)
- setting security policy, [11-2](#)
- setting to immutable, [11-1](#)
- shared BEs, [6-7](#)
- shared storage URIs, [13-1](#)
- SMF service status, [5-6](#)
- support for beadm, [6-5](#)
- time values, [3-6](#)
- transformations using archives, [7-19](#)
- troubleshooting, [12-1](#)
- updating in parallel, [6-3](#)
- usage examples, [4-1](#), [5-1](#)
- UUIDs, [12-3](#)
- zonecfg command, [9-1](#)
- zonep2vchk tool, [7-21](#)
- zonestat command, [5-7](#)
- zonestat statistics, [5-7](#)
- zones delegated restarter
 - administering, [5-5](#)
 - displaying, [5-5](#)
 - how it works, [5-5](#)
 - overview, [5-5](#)
 - properties description, [5-6](#)
 - setting properties, [5-5](#)
- zones on shared storage (ZOSS)
 - assigning shared storage, [13-5](#)
 - automated ZFS storage, [13-7](#)
 - command references, [13-27](#)
 - documentation references, [13-27](#)
 - implementing, [13-10](#)
 - migrating, [13-21](#)
 - moving to or from shared storage, [13-23](#)
 - overview, [13-1](#)
 - supported renaming methods, [13-7](#)
 - unavailable zone state and, [12-3](#), [13-8](#)
 - zoneadm subcommand options, [13-8](#)
- zonestat command
 - active monitoring, [7-7](#)
 - network bandwidth, [5-7](#)
 - reporting resource usage, [5-7](#)
- zonestatd daemon
 - monitoring resource use, [7-7](#)
- ZOSS, [13-1](#)

zpool resource, [13-6](#)
configuration considerations on shared
storage, [13-10](#)

zpool resource (*continued*)
zsched process, [6-5](#)