## Contents

Preface ...................................................................................................................................................27

Solaris Virtualization Product Overview .............................................................................................33

Part I Resource Management ..................................................................................................................35

1 Introduction to Solaris Resource Management ....................................................................................37
   Resource Management Overview ..........................................................................................................37
   Resource Classifications .......................................................................................................................38
   Resource Management Control Mechanisms .......................................................................................39
   Resource Management Configuration ....................................................................................................40
   Interaction With Non-Global Zones ......................................................................................................40
   When to Use Resource Management ..................................................................................................40
   Server Consolidation ............................................................................................................................41
   Supporting a Large or Varied User Population ....................................................................................41
   Setting Up Resource Management (Task Map) ....................................................................................42

2 Projects and Tasks (Overview) ..............................................................................................................45
   Project and Task Facilities ....................................................................................................................45
   Project Identifiers ................................................................................................................................46
   Determining a User's Default Project ....................................................................................................46
   Setting User Attributes With the useradd, usermod, and passmgmt Commands ................................47
   project Database ...............................................................................................................................47
   PAM Subsystem ..................................................................................................................................48
   Naming Services Configuration ..........................................................................................................48
   Local/etc/project File Format .............................................................................................................48
   Project Configuration for NIS ..............................................................................................................50
### Contents

Project Configuration for LDAP .............................................................. 51
Task Identifiers ...................................................................................... 51
Commands Used With Projects and Tasks ........................................... 52

3 Administering Projects and Tasks ......................................................... 55
Administering Projects and Tasks (Task Map) ....................................... 55
Example Commands and Command Options ......................................... 56
Command Options Used With Projects and Tasks ................................. 56
Using cron and su With Projects and Tasks ............................................ 58
Administering Projects ........................................................................ 59

- How to Define a Project and View the Current Project ....................... 59
- How to Delete a Project From the /etc/project File ............................. 62
- How to Validate the Contents of the /etc/project File ....................... 63
- How to Obtain Project Membership Information ............................... 63
- How to Create a New Task .............................................................. 63
- How to Move a Running Process Into a New Task ............................. 63

Editing and Validating Project Attributes .............................................. 64

- How to Add Attributes and Attribute Values to Projects .................... 64
- How to Remove Attribute Values From Projects ............................... 65
- How to Remove a Resource Control Attribute From a Project .......... 65
- How to Substitute Attributes and Attribute Values for Projects .......... 66
- How to Remove the Existing Values for a Resource Control Attribute .. 66

4 Extended Accounting (Overview) ......................................................... 67
Introduction to Extended Accounting .................................................... 67
How Extended Accounting Works ....................................................... 68
Extensible Format ................................................................................ 69
exacct Records and Format ................................................................. 69
Using Extended Accounting on a Solaris System with Zones Installed .... 70
Extended Accounting Configuration .................................................... 70
Starting and Persistently Enabling Extended Accounting ....................... 70
Records ............................................................................................... 71
Commands Used With Extended Accounting ........................................ 71
Perl Interface to l ibexacct ..................................................................... 72
5 Administering Extended Accounting (Tasks) ................................................................. 75
   Administering the Extended Accounting Facility (Task Map) ........................................ 75
   Using Extended Accounting Functionality .................................................................... 76
   ▼ How to Activate Extended Accounting for Flows, Processes, Tasks, and Network
      Componentss .............................................................................................................. 76
   How to Display Extended Accounting Status ................................................................ 77
   How to View Available Accounting Resources ............................................................. 77
   ▼ How to Deactivate Process, Task, Flow, and Network Management Accounting .......... 78
   Using the Perl Interface to libexacct ............................................................................. 79
   How to Recursively Print the Contents of an exactt Object ............................................ 79
   How to Create a New Group Record and Write It to a File ............................................ 81
   How to Print the Contents of an exactc File .................................................................... 81
   Example Output From Sun::Solaris::Exacct::Object->dump() .................................. 82

6 Resource Controls (Overview) ....................................................................................... 83
   Resource Controls Concepts ............................................................................................. 83
   Resource Limits and Resource Controls ....................................................................... 84
   Interprocess Communication and Resource Controls .................................................. 84
   Resource Control Constraint Mechanisms .................................................................. 84
   Project Attribute Mechanisms ..................................................................................... 85
   Configuring Resource Controls and Attributes ............................................................. 85
   Available Resource Controls ....................................................................................... 86
   Zone-Wide Resource Controls ...................................................................................... 89
   Units Support .................................................................................................................. 90
   Resource Control Values and Privilege Levels ............................................................. 91
   Global and Local Actions on Resource Control Values ................................................ 92
   Resource Control Flags and Properties ........................................................................ 94
   Resource Control Enforcement ..................................................................................... 95
   Global Monitoring of Resource Control Events ........................................................... 95
   Applying Resource Controls .......................................................................................... 96
   Temporarily Updating Resource Control Values on a Running System ....................... 96
   Updating Logging Status ............................................................................................... 96
   Updating Resource Controls ........................................................................................ 96
   Commands Used With Resource Controls ................................................................... 97
7 Administering Resource Controls (Tasks) ................................................................................................................. 99
   Administering Resource Controls (Task Map) ................................................................................................................. 99
   Setting Resource Controls ........................................................................................................................................... 100
   ▼ How to Set the Maximum Number of LWPs for Each Task in a Project .................................................. 100
   ▼ How to Set Multiple Controls on a Project .................................................................................................................. 101
   Using the prctl Command ........................................................................................................................................... 102
   ▼ How to Use the prctl Command to Display Default Resource Control Values ........................................... 103
   ▼ How to Use the prctl Command to Display Information for a Given Resource Control ................................. 105
   ▼ How to Use prctl to Temporarily Change a Value ..................................................................................................... 105
   ▼ How to Use prctl to Lower a Resource Control Value ............................................................................................. 106
   ▼ How to Use prctl to Display, Replace, and Verify the Value of a Control on a Project ................................ 106
   Using rctladm ................................................................................................................................................................. 106
   ▼ How to Use rctladm .................................................................................................................................................... 106
   Using ipcs .......................................................................................................................................................................... 107
   ▼ How to Use ipcs ............................................................................................................................................................. 107
   Capacity Warnings ........................................................................................................................................................... 108
   ▼ How to Determine Whether a Web Server Is Allocated Enough CPU Capacity ........................................... 108

8 Fair Share Scheduler (Overview) ........................................................................................................................................ 109
   Introduction to the Scheduler ........................................................................................................................................ 109
   CPU Share Definition ....................................................................................................................................................... 110
   CPU Shares and Process State ....................................................................................................................................... 111
   CPU Share Versus Utilization ....................................................................................................................................... 111
   CPU Share Examples ........................................................................................................................................................... 111
   Example 1: Two CPU-Bound Processes in Each Project ............................................................................................. 112
   Example 2: No Competition Between Projects ........................................................................................................... 112
   Example 3: One Project Unable to Run ........................................................................................................................... 113
   FSS Configuration ............................................................................................................................................................... 113
   Projects and Users ............................................................................................................................................................... 113
   CPU Shares Configuration ............................................................................................................................................... 114
   FSS and Processor Sets ..................................................................................................................................................... 115
   FSS and Processor Sets Examples ................................................................................................................................. 115
   Combining FSS With Other Scheduling Classes ........................................................................................................... 117
   Setting the Scheduling Class for the System ................................................................................................................... 118
   Scheduling Class on a System with Zones Installed ................................................................................................. 118
### 9 Administrating the Fair Share Scheduler (Tasks)

- Administering the Fair Share Scheduler (Task Map) ......................................................... 119
- Monitoring the FSS ............................................................................................................. 120
  - ▼ How to Monitor System CPU Usage by Projects ......................................................... 120
  - ▼ How to Monitor CPU Usage by Projects in Processor Sets ......................................... 120
- Configuring the FSS ............................................................................................................ 120
  - ▼ How to Make FSS the Default Scheduler Class ......................................................... 121
  - ▼ How to Manually Move Processes From the TS Class Into the FSS Class .................. 121
  - ▼ How to Manually Move Processes From All User Classes Into the FSS Class .......... 122
  - ▼ How to Manually Move a Project's Processes Into the FSS Class ............................. 122
  - How to Tune Scheduler Parameters ............................................................................... 122

### 10 Physical Memory Control Using the Resource Capping Daemon (Overview)

- Introduction to the Resource Capping Daemon ................................................................. 125
- How Resource Capping Works .......................................................................................... 126
- Attribute to Limit Physical Memory Usage for Projects ................................................... 126
  - rcapd Configuration ....................................................................................................... 127
    - Using the Resource Capping Daemon on a System With Zones Installed ................ 127
    - Memory Cap Enforcement Threshold .......................................................................... 128
    - Determining Cap Values ............................................................................................. 128
    - rcapd Operation Intervals .......................................................................................... 130
- Monitoring Resource Utilization With rcapstat ............................................................... 131
- Commands Used With rcapd .............................................................................................. 132

### 11 Administrating the Resource Capping Daemon (Tasks)

- Setting the Resident Set Size Cap ..................................................................................... 133
  - ▼ How to Add an rcap.max-rss Attribute for a Project ................................................ 133
  - ▼ How to Use the projmod Command to Add an rcap.max-rss Attribute for a Project .... 134
- Configuring and Using the Resource Capping Daemon (Task Map) ................................. 134
  - Administering the Resource Capping Daemon With rcapadm ..................................... 135
    - ▼ How to Set the Memory Cap Enforcement Threshold .............................................. 135
    - ▼ How to Set Operation Intervals ................................................................................. 136
Contents

12 Resource Pools (Overview) ................................................................. 143

Introduction to Resource Pools ........................................................... 144
Introduction to Dynamic Resource Pools .............................................. 145
About Enabling and Disabling Resource Pools and Dynamic Resource Pools 145
Resource Pools Used in Zones ............................................................ 145
When to Use Pools ............................................................................ 146
Resource Pools Framework ................................................................. 147
/etc/pooladm.conf Contents .............................................................. 148
Pools Properties ................................................................................ 148
Implementing Pools on a System ......................................................... 149
/project.pool Attribute ..................................................................... 149
SPARC: Dynamic Reconfiguration Operations and Resource Pools .... 149
Creating Pools Configurations .......................................................... 150
Directly Manipulating the Dynamic Configuration ......................... 151
poold Overview ................................................................................. 151
Managing Dynamic Resource Pools .................................................. 151
Configuration Constraints and Objectives ........................................... 152
Configuration Constraints ............................................................... 152
Configuration Objectives ................................................................ 153
poold Properties .............................................................................. 155
poold Features That Can Be Configured ............................................ 156
poold Monitoring Interval ............................................................... 157
poold Logging Information .............................................................. 157
Logging Location ............................................................................. 159
Log Management With logadm ....................................................... 159
How Dynamic Resource Allocation Works .................................... 159
About Available Resources ........................................................................................................ 159
Determining Available Resources ............................................................................................ 160
Identifying a Resource Shortage ............................................................................................... 160
Determining Resource Utilization ........................................................................................... 161
Identifying Control Violations ................................................................................................. 161
Determining Appropriate Remedial Action ........................................................................... 161
Using poolstat to Monitor the Pools Facility and Resource Utilization .............................. 162
poolstat Output ....................................................................................................................... 162
Tuning poolstat Operation Intervals .................................................................................... 163
Commands Used With the Resource Pools Facility ............................................................... 163

13 Creating and Administering Resource Pools (Tasks) .................................................... 165
Administering Resource Pools (Task Map) ........................................................................... 165
Enabling and Disabling the Pools Facility ............................................................................. 167
  ▼ How to Enable the Resource Pools Service Using svcadm ............................................. 167
  ▼ How to Disable the Resource Pools Service Using svcadm ......................................... 167
  ▼ How to Enable the Dynamic Resource Pools Service Using svcadm ............................ 168
  ▼ How to Disable the Dynamic Resource Pools Service Using svcadm .......................... 170
  ▼ How to Enable Resource Pools Using pooladm .............................................................. 171
  ▼ How to Disable Resource Pools Using pooladm ............................................................. 171
Configuring Pools ................................................................................................................ 171
  ▼ How to Create a Static Configuration ............................................................................ 171
  ▼ How to Modify a Configuration ....................................................................................... 173
  ▼ How to Associate a Pool With a Scheduling Class ......................................................... 175
  ▼ How to Set Configuration Constraints .......................................................................... 177
  ▼ How to Define Configuration Objectives ........................................................................ 178
  ▼ How to Set the poold Logging Level ............................................................................. 180
  ▼ How to Use Command Files With poolcfg .................................................................... 181
Transferring Resources ........................................................................................................... 182
  ▼ How to Move CPUs Between Processor Sets ................................................................. 182
Activating and Removing Pool Configurations ...................................................................... 182
  ▼ How to Activate a Pools Configuration ......................................................................... 183
  ▼ How to Validate a Configuration Before Committing the Configuration .......................... 183
  ▼ How to Remove a Pools Configuration ......................................................................... 183
Setting Pool Attributes and Binding to a Pool .................................................................... 184
17 Planning and Configuring Non-Global Zones (Tasks) ................................................................. 241
Planning and Configuring a Non-Global Zone (Task Map) ............................................................. 241
Evaluating the Current System Setup ............................................................................................... 244
Disk Space Requirements ................................................................................................................ 244
Restricting Zone Size .................................................................................................................... 245
Determine the Zone Host Name and the Network Requirements .................................................... 246
Zone Host Name .............................................................................................................................. 246
Shared-IP Zone Network Address ................................................................................................... 246
Exclusive-IP Zone Network Address ............................................................................................... 247
File System Configuration ................................................................................................................ 247
Creating, Revising, and Deleting Non-Global Zone Configurations (Task Map) ............. 248
Configuring, Verifying, and Committing a Zone ................................................................. 249
  ▼ How to Configure the Zone ............................................................................................. 249
  Where to Go From Here ..................................................................................................... 254
  Script to Configure Multiple Zones .................................................................................. 255
  ▼ How to Display the Configuration of a Non-Global Zone .......................................... 257
Using the zonecfg Command to Modify a Zone Configuration ........................................ 257
  ▼ How to Modify a Resource Type in a Zone Configuration ............................................. 257
  ▼ How to Clear a Property Type in a Zone Configuration ............................................... 258
  ▼ How to Rename a Zone .................................................................................................. 259
  ▼ How to Add a Dedicated Device to a Zone ..................................................................... 260
  ▼ How to Set zone.cpu-shares in the Global Zone ......................................................... 260
Using the zonecfg Command to Revert or Remove a Zone Configuration .......................... 261
  ▼ How to Revert a Zone Configuration ............................................................................ 261
  ▼ How to Delete a Zone Configuration .............................................................................. 263

18 About Installing, Halting, Uninstalling, and Cloning Non-Global Zones (Overview) .... 265
Zone Installation and Administration Concepts ..................................................................... 265
Zone Construction .................................................................................................................. 266
The zoneadm Daemon ............................................................................................................ 268
The zsched Zone Scheduler .................................................................................................. 268
Zone Application Environment .............................................................................................. 268
About Halting, Rebooting, and Uninstalling Zones............................................................... 269
  Halting a Zone .................................................................................................................. 269
  Rebooting a Zone .............................................................................................................. 269
  Zone Boot Arguments ....................................................................................................... 269
  Zone autoboot .................................................................................................................... 270
  Uninstalling a Zone ........................................................................................................... 270
About Cloning Non-Global Zones ........................................................................................ 271

19 Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks) .... 273
Zone Installation (Task Map) ................................................................................................. 273
Installing and Booting Zones ................................................................................................. 274
  ▼ (Optional) How to Verify a Configured Zone Before It Is Installed ............................... 274
  ▼ How to Install a Configured Zone ................................................................................ 275
Non-Global Zones as NFS Clients ................................................................. 358
Use of mknod Prohibited in a Zone .............................................................. 358
Traversing File Systems .............................................................................. 358
Restriction on Accessing A Non-Global Zone From the Global Zone .......... 359
Networking in Shared-IP Non-Global Zones ............................................. 360
  Shared-IP Zone Partitioning ................................................................. 360
  Shared-IP Network Interfaces .............................................................. 361
  IP Traffic Between Shared-IP Zones on the Same Machine .................. 361
  Solaris IP Filter in Shared-IP Zones ....................................................... 362
  IP Network Multipathing in Shared-IP Zones ........................................ 362
Networking in Exclusive-IP Non-Global Zones ........................................... 362
  Exclusive-IP Zone Partitioning ............................................................. 363
  Exclusive-IP Data-Link Interfaces ......................................................... 363
  IP Traffic Between Exclusive-IP Zones on the Same Machine .............. 363
  Solaris IP Filter in Exclusive-IP Zones .................................................. 363
  IP Network Multipathing in Exclusive-IP Zones .................................... 363
Device Use in Non-Global Zones ............................................................... 364
  /dev and the /devices Namespace ......................................................... 364
  Exclusive-Use Devices ........................................................................ 364
  Device Driver Administration .............................................................. 364
  Utilities That Do Not Work or Are Modified in Non-Global Zones ......... 365
Running Applications in Non-Global Zones .............................................. 365
Resource Controls Used in Non-Global Zones .......................................... 366
Fair Share Scheduler on a Solaris System With Zones Installed ............... 366
  FSS Share Division in a Global or Non-Global Zone ......................... 366
  Share Balance Between Zones ........................................................... 367
Extended Accounting on a Solaris System With Zones Installed ............. 367
Privileges in a Non-Global Zone ............................................................... 367
Using IP Security Architecture in Zones .................................................. 371
  IP Security Architecture in Shared-IP Zones ....................................... 371
  IP Security Architecture in Exclusive-IP Zones .................................... 371
Using Solaris Auditing in Zones .............................................................. 372
  Configuring Audit in the Global Zone .................................................. 372
  Configuring User Audit Characteristics in a Non-Global Zone .......... 373
  Providing Audit Records for a Specific Non-Global Zone ................. 373
Core Files in Zones .................................................................................. 373
Running DTrace in a Non-Global Zone ................................................................. 373
About Backing Up a Solaris System With Zones Installed ............................... 374
  Backing Up Loopback File System Directories ............................................... 374
  Backing Up Your System From the Global Zone ............................................. 374
  Backing Up Individual Non-Global Zones on Your System ......................... 375
Determining What to Back Up in Non-Global Zones ........................................ 375
  Backing Up Application Data Only ................................................................. 375
  General Database Backup Operations .......................................................... 376
  Tape Backups .................................................................................................. 376
About Restoring Non-Global Zones .................................................................... 376
Commands Used on a Solaris System With Zones Installed ............................. 377

27 Administering Solaris Zones (Tasks) ................................................................. 383
  Using the ppriv Utility ..................................................................................... 383
    ▼ How to List Solaris Privileges in the Global Zone ...................................... 383
    ▼ How to List the Non-Global Zone's Privilege Set ...................................... 384
    ▼ How to List a Non-Global Zone's Privilege Set With Verbose Output .......... 384
  Using DTrace in a Non-Global Zone ............................................................... 385
    ▼ How to Use DTrace .................................................................................... 385
  Checking the Status of SMF Services in a Non-Global Zone ......................... 386
    ▼ How to Check the Status of SMF Services From the Command Line .......... 386
    ▼ How to Check the Status of SMF Services From Within a Zone ................ 386
  Mounting File Systems in Running Non-Global Zones .................................... 387
    ▼ SX Only: How to Import Raw and Block Devices by Using zonecfg .......... 387
    ▼ How to Mount the File System Manually ................................................. 388
    ▼ How to Place a File System in /etc/vfstab to Be Mounted When the Zone Boots .... 389
    ▼ How to Mount a File System From the Global Zone Into a Non-Global Zone ........ 389
  Adding Non-Global Zone Access to Specific File Systems in the Global Zone .... 390
    ▼ How to Add Access to CD or DVD Media in a Non-Global Zone .............. 390
    ▼ How to Add a Writable Directory under /usr in a Non-Global Zone .......... 392
    ▼ How to Export Home Directories in the Global Zone Into a Non-Global Zone .......... 393
  Using IP Network Multipathing on a Solaris System With Zones Installed .... 393
    ▼ How to Use IP Network Multipathing in Exclusive-IP Non-Global Zones .... 393
    ▼ How to Extend IP Network Multipathing Functionality to Shared-IP Non-Global Zones ......
Administering Data-Links in Exclusive-IP Non-Global Zones ................................................... 395
  ▼ How to Use dladm show-linkprop ................................................................. 395
  ▼ How to Use dladm set-linkprop ................................................................. 396
  ▼ How to Use dladm reset-linkprop ............................................................ 397
Using the Fair Share Scheduler on a Solaris System With Zones Installed ......................... 397
  ▼ How to Set FSS Shares in the Global Zone Using the prctl Command .................. 397
  ▼ How to Change the zone.cpu-shares Value in a Zone Dynamically .................. 398
Using Rights Profiles in Zone Administration ................................................................. 398
  ▼ How to Assign the Zone Management Profile ............................................. 398
  Example—Using Profile Shells With Zone Commands ......................................... 399
Backing Up a Solaris System With Installed Zones ......................................................... 399
  ▼ How to Use ufsdump to Perform Backups .................................................. 399
  ▼ How to Create a UFS Snapshot Using fssnap ............................................. 400
  ▼ How to Use find and cpio to Perform Backups ........................................... 401
  ▼ How to Print a Copy of a Zone Configuration ............................................. 402
Restoring a Non-Global Zone ......................................................................................... 402
  ▼ How to Restore an Individual Non-Global Zone ......................................... 402

28 Troubleshooting Miscellaneous Solaris Zones Problems .................................................. 405
Information on Zones in the OpenSolaris 2009.06 Release .............................................. 405
Exclusive-IP Zone Is Using Device, so dladm reset-linkprop Fails .................................. 405
Incorrect Privilege Set Specified in Zone Configuration .................................................. 405
Zone Administrator Mounting Over File Systems Populated by the Global Zone ............ 406
netmasks Warning Displayed When Booting Zone ...................................................... 406
Zone Does Not Halt ................................................................................................. 407
Resolving Problems With a zoneadm attach Operation ................................................. 407
  ▼ Patches and Packages Are Out of Sync .................................................... 407
  ▼ Operating System Releases Do Not Match .............................................. 408
  ▼ Machine Architectures Do Not Match ...................................................... 409

Part III Linux Branded Zones .......................................................................................... 411

29 About Branded Zones and the Linux Branded Zone ....................................................... 413
About Using Zones on a Solaris System ......................................................................... 413
30 Planning the lx Branded Zone Configuration (Overview) ................................................................. 421

System and Space Requirements .................................................................................................... 421
Restricting the Size of the Branded Zone ...................................................................................... 421
Branded Zone Network Address .................................................................................................. 422
lx Branded Zone Configuration Process ......................................................................................... 422
lx Branded Zone Configuration Components .................................................................................. 422

Zone Name and Zone Path in an lx Branded Zone .......................................................................... 422
Zone Autoboot in an lx Branded Zone .............................................................................................. 423
Resource Pool Association in an lx Branded Zone ......................................................................... 423
Specifying the dedicated-cpu Resource ............................................................................................ 423
Specifying the capped-cpu Resource ............................................................................................... 424
Scheduling Class in a Zone ................................................................................................................ 424
capped-memory Resource ............................................................................................................... 425
Zone Network Interfaces in an lx Branded Zone .......................................................................... 425
Mounted File Systems in an lx Branded Zone ................................................................................. 426
Zone-Wide Resource Controls in an lx Branded Zone ................................................................. 426
Configurable Privileges in an lx Branded Zone ............................................................................. 428
attr Resource in an lx Branded Zone .............................................................................................. 428
Resources Included in the Configuration by Default ..................................................................... 429
Configured Devices in lx Branded Zones ....................................................................................... 429
File Systems Defined in lx Branded Zones ..................................................................................... 429
Privileges Defined in lx Branded Zones .......................................................................................... 429
Using the zonecfg Command to Create an lx Branded Zone ......................................................... 430
31 Configuring the \texttt{lx} Branded Zone (Tasks) ........................................................................................................ 439
Planning and Configuring an \texttt{lx} Branded Zone (Task Map) ................................................................ 439
How to Configure the \texttt{lx} Branded Zone ........................................................................................................ 440
\hline
How to Configure, Verify, and Commit the \texttt{lx} Branded Zone ........................................................................ 441
Where to Go From Here ..................................................................................................................................... 445
Script to Configure Multiple \texttt{lx} Branded Zones .......................................................................................... 445
\hline
How to Display the Configuration of a Branded Zone .................................................................................. 447
Modifying, Reverting, or Removing Zone Configurations ............................................................................... 447

32 About Installing, Booting, Halting, Cloning, and Uninstalling \texttt{lx} Branded Zones (Overview) ........................................ 449
Branded Zone Installation and Administration Concepts .............................................................................. 449
\texttt{lx} Branded Zone Installation Methods ................................................................................................ 450
\texttt{lx} Branded Zone Construction ............................................................................................................... 451
zoneadmd Zones Administration Daemon ...................................................................................................... 451
zsched Zone Scheduling Process ................................................................................................................ 451
Branded Zone Application Environment ..................................................................................................... 451
Passwords .......................................................................................................................................................... 452
About Halting, Rebooting, Uninstalling, and Cloning \texttt{lx} Branded Zones ....................................................... 452
Halting a Branded Zone .................................................................................................................................. 452
Rebooting a Branded Zone ............................................................................................................................. 452
Branded Zone Boot Arguments .................................................................................................................... 452
Branded Zone \texttt{autoboot} ............................................................................................................................. 453
Uninstalling the Branded Zone ...................................................................................................................... 453
About Cloning an \texttt{lx} Branded Zone ........................................................................................................... 453
Booting and Rebooting \texttt{lx} Branded Zones ................................................................................................. 453
33 Installing, Booting, Halting, Uninstalling and Cloning lx Branded Zones (Tasks) .................455
   lx Branded Zone Installation (Task Map) ...........................................................................455
   Installing and Booting lx Branded Zones ........................................................................456
      ▼ How to Obtain the Linux Archives .............................................................................456
      ▼ How to Install an lx Branded Zone .........................................................................456
      ▼ How to Install a Subset of the Packages ..................................................................459
      ▼ How to Enable Networking in an lx Branded Zone ..................................................459
      ▼ How to Obtain the UUID of an Installed Branded Zone .............................................459
      ▼ How to Mark an Installed lx Branded Zone Incomplete .............................................460
         (Optional) Placing an Installed lx Branded Zone in the Ready State .........................461
      ▼ How to Boot an lx Branded Zone ..............................................................................461
      ▼ How to Boot an lx Branded Zone in Single-User Mode .............................................462
         Where to Go From Here ...............................................................................................462
   Halting, Rebooting, Uninstalling, Cloning, and Deleting lx Branded Zones (Task Map) ....463
      Halting, Rebooting, and Uninstalling lx Branded Zones .............................................463
         Cloning an lx Branded Zone on the Same System ......................................................466
            ▼ How to Clone an lx Branded Zone .......................................................................466
            ▼ How to Clone a Zone from an Existing Snapshot ...............................................467
            ▼ How to Use Copy Instead of ZFS Clone ...............................................................468
         Deleting an lx Branded Zone From the System .........................................................469
            ▼ How to Remove an lx Branded Zone ...................................................................469

34 Logging In to lx Branded Zones (Tasks) ...........................................................................471
   zlogin Command Overview ..............................................................................................471
   lx Branded Zone Login Methods ......................................................................................472
   Login Procedures for Branded Zones (Task Map) ..............................................................472
   Logging In to an lx Branded Zone .....................................................................................472
      ▼ How to Log In to the lx Branded Zone Console .........................................................473
      ▼ How to Use Interactive Mode to Access a Branded Zone ..........................................473
      ▼ How to Verify the Running Environment ..................................................................474
      ▼ How to Use Non-Interactive Mode to Access an lx Branded Zone .............................474
      ▼ How to Exit the lx Branded Zone ..............................................................................475
      ▼ How to Use Failsafe Mode to Enter an lx Branded Zone ...........................................475
      ▼ How to Use zlogin to Shut Down an lx Branded Zone .............................................476
Verifying System Elements ........................................................................................................ .... 548
▼ How to Verify Devices in /dev/xen ....................................................................................... 548
▼ How to Verify That the xvm Hypervisor Services Are Started .............................................. 548
Tracing the Interaction Between dom0 and the Hypervisor .......................................................... 549
Guest Issues ........................................................................................................................................ 549
   Known OpenSolaris 2009.06 Guest Issues .............................................................................. 549
   Known Issues With Running Solaris 10 Guests on Other Operating Systems ................... 550
For More Information .................................................................................................................. 550

Glossary .............................................................................................................................................. 551

Index ................................................................................................................................................... 555
Preface

*System Administration Guide: Virtualization Using the Solaris Operating System* is part of a multivolume set that covers a significant part of the Solaris™ Operating System administration information. This book assumes that you have already installed the operating system and set up any networking software that you plan to use.

**Note** – This Solaris release supports systems that use the SPARC® and x86 families of processor architectures: UltraSPARC®, SPARC64, AMD64, Pentium, and Xeon EM64T. The supported systems appear in the *Solaris 10 Hardware Compatibility List* at [http://www.sun.com/bigadmin/hcl](http://www.sun.com/bigadmin/hcl). This document cites any implementation differences between the platform types.

### About the Sun xVM Hypervisor

The Sun™ xVM Hypervisor is based on the work of the Xen community. The hypervisor supports multiple operating system instances simultaneously. In a running system, the hypervisor fits between the hardware and the operating system. The hypervisor virtualizes the system’s hardware to transparently share and partition the system’s resources, such as CPUs, memory, and network interface cards (NICs), among the user domains.

### About Solaris Containers

A Solaris Container is a complete runtime environment for applications. Solaris Resource Manager and Solaris Zones software partitioning technology form the container. These components address different qualities the container can deliver and work together to create the container. The zones portion of the container provides a virtual mapping from the application to the platform resources. Zones allow application components to be isolated from one another even though the zones share a single instance of the Solaris Operating System. Resource management features permit you to allocate the quantity of resources that a workload receives.

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

See “About Zones in the OpenSolaris 2009.06 Release” on page 200 to see the differences between zones in the OpenSolaris 2009.06 release and zones in Solaris Express (SX) releases.
About Solaris Containers for Linux Applications

Solaris Containers for Linux Applications use Sun's BrandZ technology to run Linux applications on the Solaris Operating System. Linux applications run unmodified in the secure environment provided by the non-global zone feature. This enables you to use the Solaris system to develop, test, and deploy Linux applications.

To use this feature, see Part III.

About Using Solaris Zones on a Solaris Trusted Extensions System

For information on using zones on a Solaris Trusted Extensions system, see Chapter 16, "Managing Zones in Trusted Extensions (Tasks)," in Solaris Trusted Extensions Administrator's Procedures. Note that only the labeled brand can be booted on an OpenSolaris system configuration.

Who Should Use This Book

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

How the System Administration Guides Are Organized

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

<table>
<thead>
<tr>
<th>Book Title</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Administration Guide: Basic Administration</td>
<td>User accounts and groups, server and client support, shutting down and booting a system, managing services, and managing software (packages and patches)</td>
</tr>
<tr>
<td>System Administration Guide: Advanced Administration</td>
<td>Terminals and modems, system resources (disk quotas, accounting, and crontabs), system processes, and troubleshooting Solaris software problems</td>
</tr>
<tr>
<td>System Administration Guide: Devices and File Systems</td>
<td>Removable media, disks and devices, file systems, and backing up and restoring data</td>
</tr>
<tr>
<td>System Administration Guide: IP Services</td>
<td>TCP/IP network administration, IPv4 and IPv6 address administration, DHCP, IPSec, IKE, Solaris IP filter, Mobile IP, IP network multipathing (IPMP), and IPQoS</td>
</tr>
<tr>
<td>Book Title</td>
<td>Topics</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><em>System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)</em></td>
<td>DNS, NIS, and LDAP naming and directory services, including transitioning from NIS to LDAP and transitioning from NIS+ to LDAP</td>
</tr>
<tr>
<td><em>System Administration Guide: Network Interfaces and Network Virtualization</em></td>
<td>Networking stack, NIC driver property configuration, network interface configuration, administration of VLANs and link aggregations, configuring WiFi wireless networking</td>
</tr>
<tr>
<td><em>System Administration Guide: Network Services</em></td>
<td>Web cache servers, time-related services, network file systems (NFS and Autofs), mail, SLP, and PPP</td>
</tr>
<tr>
<td><em>System Administration Guide: Solaris Printing</em></td>
<td>Solaris printing topics and tasks, using services, tools, protocols, and technologies to set up and administer printing services and printers</td>
</tr>
<tr>
<td><em>System Administration Guide: Security Services</em></td>
<td>Auditing, device management, file security, BART, Kerberos services, PAM, Solaris Cryptographic Framework, privileges, RBAC, SASL, and Solaris Secure Shell</td>
</tr>
<tr>
<td><em>System Administration Guide: Virtualization Using the Solaris Operating System</em></td>
<td>Resource management features, which enable you to control how applications use available system resources; zones software partitioning technology, which virtualizes operating system services to create an isolated environment for running applications; and virtualization using Sun xVM hypervisor technology, which supports multiple operation system instances simultaneously</td>
</tr>
<tr>
<td><em>Solaris CIFS Administration Guide</em></td>
<td>Solaris CIFS service, which enables you to configure a Solaris system to make CIFS shares available to CIFS clients; and native identity mapping services, which enables you to map user and group identities between Solaris systems and Windows systems</td>
</tr>
<tr>
<td><em>Solaris Trusted Extensions Administrator’s Procedures</em></td>
<td>System installation, configuration, and administration that is specific to Solaris Trusted Extensions</td>
</tr>
<tr>
<td><em>Solaris ZFS Administration Guide</em></td>
<td>ZFS™ storage pool and file system creation and management, snapshots, clones, backups, using access control lists (ACLs) to protect ZFS files, using ZFS on a Solaris system with zones installed, emulated volumes, and troubleshooting and data recovery</td>
</tr>
</tbody>
</table>

**Related Book**

*Solaris Containers: Resource Management and Solaris Zones Developer’s Guide* describes how to write applications that partition and manage system resources and discusses which APIs to use. Programming examples and a discussion of programming issues to consider when writing an application are also provided.
Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:
- Documentation (http://www.sun.com/documentation/)
- Support (http://www.sun.com/support/)
- Training (http://www.sun.com/training/)

Typographic Conventions

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

<table>
<thead>
<tr>
<th>Typeface or Symbol</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AaBbCc123</td>
<td>The names of commands, files, and directories, and onscreen computer output</td>
<td>Edit your .login file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code></td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>What you type, contrasted with onscreen computer output</td>
<td><code>machine_name% su</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>Password:</code></td>
</tr>
<tr>
<td>aabbcc123</td>
<td>Placeholder: replace with a real name or value</td>
<td>The command to remove a file is <code>rm filename</code>.</td>
</tr>
</tbody>
</table>
### Shell Prompts in Command Examples

The table in this section shows the default UNIX system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

Note that if you have the appropriate role account on the OpenSolaris system, you can type the `pfexec` command before the privileged command:

```bash
$ pfexec acctadm -x process
```

You can execute a privileged command in a profile shell by typing `pfsh` with a Return:

```bash
$ pfsh
zoneadm -z my-zone halt
```

<table>
<thead>
<tr>
<th>Shell</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>C shell</td>
<td><code>machine_name%</code></td>
</tr>
<tr>
<td>C shell for superuser</td>
<td><code>machine_name#</code></td>
</tr>
<tr>
<td>Bourne shell and Korn shell</td>
<td><code>$</code></td>
</tr>
<tr>
<td>Bourne shell and Korn shell for superuser</td>
<td><code>#</code></td>
</tr>
</tbody>
</table>
Solaris Virtualization Product Overview

This chapter provides a brief overview of virtualization using the Solaris Operating System.

About Virtualization

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

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

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

Virtualization products offered by Sun Microsystems include the following:

- Logical Domains (LDoms), the SPARC hypervisor virtualization solution for running multiple operating system instances on a single machine simultaneously. Operating system-level virtualization features, such as zones or resource management, can be used in LDoms.
- Sun™ xVM hypervisor, the x86 hypervisor virtualization solution for running multiple operating system instances on a single machine simultaneously.
- Native and branded zones containers, which provide isolated execution environments within a Solaris operating system instance and can be run within a Solaris guest domain.
- Resource management features, which enable you to control how applications use available system resources.
- Sun VirtualBox, which allows you to run unmodified 32-bit and 64-bit operating systems as virtual machines on Intel and AMD processors, directly on your existing operating system.
Network virtualization features, used throughout virtualization technologies.

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

**Associated OpenSolaris Communities**

The OpenSolaris Xen, Zones, and Networking communities (http://www.opensolaris.org/) provide information, discussion groups, and the ability to participate in feature development for the technologies discussed in this guide. Also visit the Logical Domains community for information about LDomS technology and links to documentation.
This part covers Solaris Resource Management, which enables you to control how applications use available system resources.
Resource management functionality is a component of the Solaris™ Container environment. Resource management enables you to control how applications use available system resources. You can do the following:

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

This chapter covers the following topics.

- “Resource Management Overview” on page 37
- “When to Use Resource Management” on page 40
- “Setting Up Resource Management (Task Map)” on page 42

**Resource Management Overview**

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

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

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

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

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

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

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

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

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

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

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

**Resource Classifications**

A resource is any aspect of the computing system that can be manipulated with the intent to change application behavior. Thus, a resource is a capability that an application implicitly or explicitly requests. If the capability is denied or constrained, the execution of a robustly written application proceeds more slowly.
Classification of resources, as opposed to identification of resources, can be made along a number of axes. The axes could be implicitly requested as opposed to explicitly requested, time-based, such as CPU time, compared to time-independent, such as assigned CPU shares, and so forth.

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

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

**Resource Management Control Mechanisms**

The three types of control mechanisms that are available in the Solaris Operating System are constraints, scheduling, and partitioning.

**Constraint Mechanisms**

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

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

**Scheduling Mechanisms**

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

**Partitioning Mechanisms**

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

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

**Resource Management Configuration**

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

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

**Interaction With Non-Global Zones**

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

**When to Use Resource Management**

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

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

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

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

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

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

Supporting a Large or Varied User Population

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

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

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify the workloads on your system and categorize each workload by project.</td>
<td>Create project entries in either the /etc/project file, in the NIS map, or in the LDAP directory service.</td>
<td>&quot;project Database&quot; on page 47</td>
</tr>
<tr>
<td>Prioritize the workloads on your system.</td>
<td>Determine which applications are critical. These workloads might require preferential access to resources.</td>
<td>Refer to your business service goals.</td>
</tr>
<tr>
<td>Monitor real-time activity on your system.</td>
<td>Use performance tools to view the current resource consumption of workloads that are running on your system. You can then evaluate whether you must restrict access to a given resource or isolate particular workloads from other workloads.</td>
<td>cpustat(1M), iostat(1M), mpstat(1M), prstat(1M), sar(1), and vmstat(1M) man pages</td>
</tr>
<tr>
<td>Make temporary modifications to the workloads that are running on your system.</td>
<td>To determine which values can be altered, refer to the resource controls that are available in the Solaris system. You can update the values from the command line while the task or process is running.</td>
<td>&quot;Available Resource Controls&quot; on page 86, &quot;Global and Local Actions on Resource Control Values&quot; on page 92, &quot;Temporarily Updating Resource Control Values on a Running System&quot; on page 96 and rctladm(1M) and prctl(1) man pages</td>
</tr>
<tr>
<td>Set resource controls and project attributes for every project entry in the project database or naming service project database.</td>
<td>Each project entry in the /etc/project file or the naming service project database can contain one or more resource controls or attributes. Resource controls constrain tasks and processes attached to that project. For each threshold value that is placed on a resource control, you can associate one or more actions to be taken when that value is reached. You can set resource controls by using the command-line interface. Certain configuration parameters can also be set by using the Solaris Management Console.</td>
<td>&quot;project Database&quot; on page 47, &quot;Local /etc/project File Format&quot; on page 48, &quot;Available Resource Controls&quot; on page 86, &quot;Global and Local Actions on Resource Control Values&quot; on page 92, and Chapter 8, &quot;Fair Share Scheduler (Overview)&quot;</td>
</tr>
<tr>
<td>Place an upper bound on the resource consumption of physical memory by collections of processes attached to a project.</td>
<td>The resource cap enforcement daemon will enforce the physical memory resource cap defined for the project's rcap.max-rss attribute in the /etc/project file.</td>
<td>&quot;project Database&quot; on page 47 and Chapter 10, &quot;Physical Memory Control Using the Resource Capping Daemon (Overview)&quot;</td>
</tr>
<tr>
<td>Task</td>
<td>Description</td>
<td>For Instructions</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Create resource pool configurations.</td>
<td>Resource pools provide a way to partition system resources, such as processors, and maintain those partitions across reboots. You can add one <code>project.pool</code> attribute to each entry in the <code>/etc/project</code> file.</td>
<td>&quot;project Database&quot; on page 47 and Chapter 12, &quot;Resource Pools (Overview)&quot;</td>
</tr>
<tr>
<td>Make the fair share scheduler (FSS) your default system scheduler.</td>
<td>Ensure that all user processes in either a single CPU system or a processor set belong to the same scheduling class.</td>
<td>&quot;Configuring the FSS&quot; on page 120 and <code>dispadmin(1M)</code> man page</td>
</tr>
<tr>
<td>Activate the extended accounting facility to monitor and record resource consumption on a task or process basis.</td>
<td>Use extended accounting data to assess current resource controls and to plan capacity requirements for future workloads. Aggregate usage on a system-wide basis can be tracked. To obtain complete usage statistics for related workloads that span more than one system, the project name can be shared across several machines.</td>
<td>&quot;How to Activate Extended Accounting for Flows, Processes, Tasks, and Network Components&quot; on page 76 and <code>acctadm(1M)</code> man page</td>
</tr>
<tr>
<td>(Optional) If you need to make additional adjustments to your configuration, you can continue to alter the values from the command line. You can alter the values while the task or process is running.</td>
<td>Modifications to existing tasks can be applied on a temporary basis without restarting the project. Tune the values until you are satisfied with the performance. Then, update the current values in the <code>/etc/project</code> file or in the naming service project database.</td>
<td>&quot;Temporarily Updating Resource Control Values on a Running System&quot; on page 96 and <code>rctladm(1M)</code> and <code>prctl(1)</code> man pages</td>
</tr>
<tr>
<td>(Optional) Capture extended accounting data.</td>
<td>Write extended accounting records for active processes and active tasks. The files that are produced can be used for planning, chargeback, and billing purposes. There is also a Practical Extraction and Report Language (Perl) interface to <code>libexacct</code> that enables you to develop customized reporting and extraction scripts.</td>
<td><code>wracct(1M)</code> man page and &quot;Perl Interface to <code>libexacct</code>&quot; on page 72</td>
</tr>
</tbody>
</table>
This chapter discusses the project and task facilities of Solaris resource management. Projects and tasks are used to label workloads and separate them from one another.

The following topics are covered in this chapter:

- “Project and Task Facilities” on page 45
- “Project Identifiers” on page 46
- “Task Identifiers” on page 51
- “Commands Used With Projects and Tasks” on page 52

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

Project and Task Facilities

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

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

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

Project Identifiers

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

Determining a User's Default Project

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

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

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

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

This logic is provided by the `getdefaultproj()` library function. See the `getprojent(3PROJECT)` man page for more information.
Setting User Attributes With the `useradd`, `usermod`, and `passmgmt` Commands

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

- `passmgmt` Modify user information
- `useradd` Set default project for user
- `usermod` Modify user information

Local files can include the following:
- `/etc/group`
- `/etc/passwd`
- `/etc/project`
- `/etc/shadow`
- `/etc/user_attr`

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

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

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

**Project Database**

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

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

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

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

**Naming Services Configuration**

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

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

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

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

**Local /etc/project File Format**

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

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

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

The fields are defined as follows:

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

\textbf{projid} \hspace{1cm} The project's unique numerical ID (PROJID) within the system. The maximum value of the \textit{projid} field is \texttt{UID\_MAX} (2147483647).

\textbf{comment} \hspace{1cm} A description of the project.

\textbf{user-list} \hspace{1cm} A comma-separated list of users who are allowed in the project.

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

\textbf{group-list} \hspace{1cm} A comma-separated list of groups of users who are allowed in the project.

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

\textbf{attributes} \hspace{1cm} A semicolon-separated list of name-value pairs, such as resource controls (see Chapter 6, "Resource Controls (Overview)"). \textit{name} is an arbitrary string that specifies the object-related attribute, and \textit{value} is the optional value for that attribute.

\texttt{name [ = value ]}

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

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

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

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

This example shows the default \texttt{/etc/project} file:
This example shows the default /etc/project file with project entries added at the end:

```
system:0:System:::
user.root:1:Super-User:::
noproject:2:No Project:::
default:3:::
group.staff:10:::
user.ml:2424:Lyle Personal:::
booksite:4113:Book Auction Project:ml,mp,jtd,kjh::
```

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

- To add resource controls for a project, see “Setting Resource Controls” on page 100.
- To define a physical memory resource cap for a project using the resource capping daemon described in `rcapd(1M)`, see “Attribute to Limit Physical Memory Usage for Projects” on page 126.
- To add a project.pool attribute to a project’s entry, see “Creating the Configuration” on page 190.

**Project Configuration for NIS**

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

```
project: nis files
```

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

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

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

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

project: ldap files

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

Task Identifiers

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

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

FIGURE 2–1 Project and Task Tree

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

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

- login
- cron
- newtask
- setproject
- su
You can create a finalized task by using one of the following methods. All further attempts to create new tasks will fail.

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

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

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

### Commands Used With Projects and Tasks

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

<table>
<thead>
<tr>
<th>Man Page Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>projects(1)</code></td>
<td>Displays project memberships for users. Lists projects from <code>project</code> database. Prints information on given projects. If no project names are supplied, information is displayed for all projects. Use the <code>projects</code> command with the <code>-l</code> option to print verbose output.</td>
</tr>
<tr>
<td><code>newtask(1)</code></td>
<td>Executes the user’s default shell or specified command, placing the execution command in a new task that is owned by the specified project. <code>newtask</code> can also be used to change the task and the project binding for a running process. Use with the <code>-F</code> option to create a finalized task.</td>
</tr>
<tr>
<td><code>passmgmt(1M)</code></td>
<td>Updates information in the password files. Use with the <code>-K key=value</code> option to add to user attributes or replace user attributes in local files.</td>
</tr>
<tr>
<td><code>projadd(1M)</code></td>
<td>Adds a new project entry to the <code>/etc/project</code> file. The <code>projadd</code> command creates a project entry only on the local system. <code>projadd</code> cannot change information that is supplied by the network naming service. Can be used to edit project files other than the default file, <code>/etc/project</code>. Provides syntax checking for <code>project</code> file. Validates and edits project attributes. Supports scaled values.</td>
</tr>
</tbody>
</table>
## Commands Used With Projects and Tasks

<table>
<thead>
<tr>
<th>Man Page Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>projmod(1M)</td>
<td>Modifies information for a project on the local system. <em>projmod</em> cannot change information that is supplied by the network naming service. However, the command does verify the uniqueness of the project name and project ID against the external naming service. Can be used to edit project files other than the default file, <em>/etc/project</em>. Provides syntax checking for project file. Validates and edits project attributes. Can be used to add a new attribute, add values to an attribute, or remove an attribute. Supports scaled values. Can be used with the <code>-A</code> option to apply the resource control values found in the project database to the active project. Existing values that do not match the values defined in the project file are removed.</td>
</tr>
<tr>
<td>projdel(1M)</td>
<td>Deletes a project from the local system. <em>projdel</em> cannot change information that is supplied by the network naming service.</td>
</tr>
<tr>
<td>useradd(1M)</td>
<td>Adds default project definitions to the local files. Use with the <code>-K key=value</code> option to add or replace user attributes.</td>
</tr>
<tr>
<td>userdel(1M)</td>
<td>Deletes a user's account from the local file.</td>
</tr>
<tr>
<td>usermod(1M)</td>
<td>Modifies a user's login information on the system. Use with the <code>-K key=value</code> option to add or replace user attributes.</td>
</tr>
</tbody>
</table>
Administering Projects and Tasks

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

The following topics are covered.

- “Example Commands and Command Options” on page 56
- “Administering Projects” on page 59

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

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

Administering Projects and Tasks (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>View examples of commands and options used with projects and tasks.</td>
<td>Display task and project IDs, display various statistics for processes and projects that are currently running on your system.</td>
<td>“Example Commands and Command Options” on page 56</td>
</tr>
<tr>
<td>Define a project.</td>
<td>Add a project entry to the /etc/project file and alter values for that entry.</td>
<td>“How to Define a Project and View the Current Project” on page 59</td>
</tr>
<tr>
<td>Delete a project.</td>
<td>Remove a project entry from the /etc/project file.</td>
<td>“How to Delete a Project From the /etc/project File” on page 62</td>
</tr>
<tr>
<td>Task</td>
<td>Description</td>
<td>For Instructions</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Validate the project file or project database.</td>
<td>Check the syntax of the /etc/project file or verify the uniqueness of the project name and project ID against the external naming service.</td>
<td>“How to Validate the Contents of the /etc/project File” on page 63</td>
</tr>
<tr>
<td>Obtain project membership information.</td>
<td>Display the current project membership of the invoking process.</td>
<td>“How to Obtain Project Membership Information” on page 63</td>
</tr>
<tr>
<td>Create a new task.</td>
<td>Create a new task in a particular project by using the newtask command.</td>
<td>“How to Create a New Task” on page 63</td>
</tr>
<tr>
<td>Associate a running process with a different task and project.</td>
<td>Associate a process number with a new task ID in a specified project.</td>
<td>“How to Move a Running Process Into a New Task” on page 63</td>
</tr>
<tr>
<td>Add and work with project attributes.</td>
<td>Use the project database administration commands to add, edit, validate, and remove project attributes.</td>
<td>“Editing and Validating Project Attributes” on page 64</td>
</tr>
</tbody>
</table>

**Example Commands and Command Options**

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

**Command Options Used With Projects and Tasks**

**ps Command**

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

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

**id Command**

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

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

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

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

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

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

prstat Command

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

```
% prstat -J
```

```
PID  USERNAME  SIZE  RSS  STATE  PRI  NICE  TIME  CPU  PROCESS/NLWP
21634 jtd  5512K  4848K  cpu0  44  0  0:00.00  0.3%  prstat/1
15497 jtd  48M   41M  sleep  49  0  0:08.26  0.1%  adeptedit/1
 328 root  2856K  2600K  sleep  58  0  0:00.00  0.0%  mibiisa/11
 1979 jtd  1568K  1352K  sleep  49  0  0:00.00  0.0%  csh/1
 1977 jtd  7256K  5512K  sleep  49  0  0:00.00  0.0%  dtterm/1
 192 root  3680K  2856K  sleep  58  0  0:00.36  0.0%  automountd/5
1045 jtd   24M   22M  sleep  49  0  0:00.29  0.0%  dtmail/11
 1009 jtd  9864K  8384K  sleep  58  0  0:00.59  0.0%  dtwm/8
 114 root  1640K  704K  sleep  49  0  0:01.16  0.0%  in.routed/1
 180 daemon  2704K  1944K  sleep  58  0  0:00.00  0.0%  statd/4
 145 root  2120K  1520K  sleep  58  0  0:00.00  0.0%  ypbind/1
 181 root  1864K  1336K  sleep  51  0  0:00.00  0.0%  lockd/1
 173 root  2584K  2136K  sleep  58  0  0:00.00  0.0%  inetd/1
 135 root  2960K  1424K  sleep  58  0  0:00.00  0.0%  keyserv/4
PROJID  NPROC  SIZE  RSS  MEMORY  TIME  CPU  PROJECT
 10    52   400M  271M  68%  0:11.45  0.1%  booksite
    0    35  113M  129M  32%  0:10.46  0.0%  system
```

Total: 87 processes, 205 lwps, load averages: 0.05, 0.02, 0.02

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

```
% prstat -T
```

```
PID  USERNAME  SIZE  RSS  STATE  PRI  NICE  TIME  CPU  PROCESS/NLWP
23476 jtd  51M  45M  sleep  49  0  0:04:31  0.5%  adeptedit/1
23432 jtd  6928K  5064K  sleep  59  0  0:00:00  0.1%  dtterm/1
```
<table>
<thead>
<tr>
<th>TASKID</th>
<th>NPROC</th>
<th>SIZE</th>
<th>RSS</th>
<th>MEMORY</th>
<th>TIME</th>
<th>CPU</th>
<th>PROJECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>222</td>
<td>30</td>
<td>229M</td>
<td>161M</td>
<td>44%</td>
<td>0:05:54</td>
<td>0.6%</td>
<td>group.staff</td>
</tr>
<tr>
<td>223</td>
<td>1</td>
<td>26M</td>
<td>20M</td>
<td>5.3%</td>
<td>0:03:18</td>
<td>0.6%</td>
<td>group.staff</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>61M</td>
<td>33M</td>
<td>8.9%</td>
<td>0:00:31</td>
<td>0.0%</td>
<td>group.staff</td>
</tr>
<tr>
<td>1</td>
<td>33</td>
<td>85M</td>
<td>53M</td>
<td>14%</td>
<td>0:03:33</td>
<td>0.0%</td>
<td>system</td>
</tr>
</tbody>
</table>

Total: 65 processes, 154 lwps, load averages: 0.04, 0.05, 0.06

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

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

### cron Command

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

### su Command

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

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

```
# su user
```
Administering Projects

How to Define a Project and View the Current Project

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

1. Become superuser or assume an equivalent role.

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

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

user.root
    projid : 1
    comment: ""
    users : (none)
    groups : (none)
    attribs: 

noproject
    projid : 2
    comment: ""
    users : (none)
    groups : (none)
    attribs:

default
    projid : 3
    comment: ""
    users : (none)
    groups : (none)
    attribs:

group.staff
    projid : 10
    comment: ""
    users : (none)
    groups : (none)
    attribs:
```
3 Add a project with the name `booksite`. Assign the project to a user who is named `mark` with project ID number `4113`.

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

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

```bash
# projects -l
```

```
system
   projid : 0
   comment: ""
   users : (none)
   groups : (none)
   attribs:

user.root
   projid : 1
   comment: ""
   users : (none)
   groups : (none)
   attribs:

noproject
   projid : 2
   comment: ""
   users : (none)
   groups : (none)
   attribs:

default
   projid : 3
   comment: ""
   users : (none)
   groups : (none)
   attribs:

group.staff
   projid : 10
   comment: ""
   users : (none)
   groups : (none)
   attribs:

booksite
   projid : 4113
   comment: ""
   users : mark
   groups : (none)
   attribs:
```

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

```bash
# projmod -c ‘Book Auction Project’ booksite
```
6 View the changes in the /etc/project file.

```bash
# projects -l

system
  projid : 0
  comment: ""
  users : (none)
  groups : (none)
  attribs:

user.root
  projid : 1
  comment: ""
  users : (none)
  groups : (none)
  attribs:

noproject
  projid : 2
  comment: ""
  users : (none)
  groups : (none)
  attribs:

default
  projid : 3
  comment: ""
  users : (none)
  groups : (none)
  attribs:

group.staff
  projid : 10
  comment: ""
  users : (none)
  groups : (none)
  attribs:

booksite
  projid : 4113
  comment: "Book Auction Project"
  users : mark
  groups : (none)
  attribs:
```

See Also To bind projects, tasks, and processes to a pool, see “Setting Pool Attributes and Binding to a Pool” on page 184.
How to Delete a Project From the /etc/project File

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

1. Become superuser or assume an equivalent role.

2. Remove the project booksite by using the projdel command.
   
   # projdel booksite

3. Display the /etc/project file.

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

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

   # su - mark
   # projects
   default
How to Validate the Contents of the `/etc/project` File

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

To validate a NIS map, type the following:

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

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

```bash
# projmod -n
```

How to Obtain Project Membership Information

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

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

How to Create a New Task

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

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

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

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

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

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

   The process is now a member of the new project.

How to Move a Running Process Into a New Task

This example shows how to associate a running process with a different task and new project. To perform this action, you must either be superuser, or be the owner of the process and be a member of the new project.
1. Become superuser or assume an equivalent role.

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

2. Obtain the process ID of the `book_catalog` process.
   ```bash
   # pgrep book_catalog
   8100
   ```

3. Associate process 8100 with a new task ID in the `booksite` project.
   ```bash
   # newtask -v -p booksite -c 8100
   17
   ```
   The `-c` option specifies that `newtask` operate on the existing named process.

4. Confirm the task to process ID mapping.
   ```bash
   # pgrep -T 17
   8100
   ```

---

**Editing and Validating Project Attributes**

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

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

**How to Add Attributes and Attribute Values to Projects**

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

1. Become superuser or assume an equivalent role.

2. Add a `task.max-lwps` resource control attribute with no values in the project `myproject`. A task entering the project has only the system value for the attribute.
   ```bash
   # projmod -a -K task.max-lwps myproject
   ```
3 You can then add a value to \texttt{task.max-lwps} in the project \textit{myproject}. The value consists of a privilege level, a threshold value, and an action associated with reaching the threshold.

\texttt{\# projmod -a -K "task.max-lwps=(priv,100,deny)" myproject}

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

\texttt{\# projmod -a -K "task.max-lwps=(priv,1000,signal=KILL)" myproject}

The multiple values are separated by commas. The \texttt{task.max-lwps} entry now reads:

\texttt{task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)}

\section*{How to Remove Attribute Values From Projects}

This procedure uses the values:

\texttt{task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)}

1 \textbf{Become superuser or assume an equivalent role.}

2 To remove an attribute value from the resource control \texttt{task.max-lwps} in the project \textit{myproject}, use the \texttt{projmod} command with the -r and -K options.

\texttt{\# projmod -r -K "task.max-lwps=(priv,100,deny)" myproject}

If \texttt{task.max-lwps} has multiple values, such as:

\texttt{task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)}

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

\texttt{task.max-lwps=(priv,1000,signal=KILL)}

\section*{How to Remove a Resource Control Attribute From a Project}

To remove the resource control \texttt{task.max-lwps} in the project \textit{myproject}, use the \texttt{projmod} command with the -r and -K options.

1 \textbf{Become superuser or assume an equivalent role.}

2 \textbf{Remove the attribute} \texttt{task.max-lwps} and all of its values from the project \textit{myproject}:

\texttt{\# projmod -r -K \texttt{task.max-lwps myproject}}
How to Substitute Attributes and Attribute Values for Projects

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

1. Become superuser or assume an equivalent role.

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

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

   The result would be:

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

How to Remove the Existing Values for a Resource Control Attribute

1. Become superuser or assume an equivalent role.

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

   ```
   # projmod -s -K task.max-lwps myproject
   ```
By using the project and task facilities that are described in Chapter 2, “Projects and Tasks (Overview),” to label and separate workloads, you can monitor resource consumption by each workload. You can use the extended accounting subsystem to capture a detailed set of resource consumption statistics on both processes and tasks.

The following topics are covered in this chapter.

- “Introduction to Extended Accounting” on page 67
- “How Extended Accounting Works” on page 68
- “Extended Accounting Configuration” on page 70
- “Commands Used With Extended Accounting” on page 71
- “Perl Interface to libexacct” on page 72

To begin using extended accounting, skip to “How to Activate Extended Accounting for Flows, Processes, Tasks, and Network Components” on page 76.

**Introduction to Extended Accounting**

The extended accounting subsystem labels usage records with the project for which the work was done. You can also use extended accounting, in conjunction with the Internet Protocol Quality of Service (IPQoS) flow accounting module described in Chapter 31, “Using Flow Accounting and Statistics Gathering (Tasks),” in System Administration Guide: IP Services, to capture network flow information on a system.

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

- Tasks.
- Processes.
Selectors provided by the IPQoS flowacct module. For more information, see ipqos(7IPP).

Network management. See dladm(1M) and flowadm(1M).

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

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

How Extended Accounting Works

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

If extended accounting is active, statistics are gathered that can be examined by the libexacct API. libexacct allows examination of the exactc files either forward or backward. The API supports third-party files that are generated by libexacct as well as those files that are created by the kernel. There is a Practical Extraction and Report Language (Perl) interface to libexacct that enables you to develop customized reporting and extraction scripts. See “Perl Interface to libexacct” on page 72.

For example, with extended accounting enabled, the task tracks the aggregate resource usage of its member processes. A task accounting record is written at task completion. Interim records on running processes and tasks can also be written. For more information on tasks, see Chapter 2, “Projects and Tasks (Overview).”
The extended accounting format is substantially more extensible than the SunOS™ legacy system accounting software format (see “What is System Accounting?” in System Administration Guide: Advanced Administration). Extended accounting permits accounting metrics to be added and removed from the system between releases, and even during system operation.

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

**exacct Records and Format**

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

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

**Note** – The putacct system call is also available from the Perl interface.
The format permits different forms of accounting records to be captured without requiring that every change be an explicit version change. Well-written applications that consume accounting data must ignore records they do not understand.

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

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

Using Extended Accounting on a Solaris System with Zones Installed

The extended accounting subsystem collects and reports information for the entire system (including non-global zones) when run in the global zone. The global administrator can also determine resource consumption on a per-zone basis. See "Extended Accounting on a Solaris System With Zones Installed" on page 367 for more information.

Extended Accounting Configuration

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

Starting and Persistently Enabling Extended Accounting

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

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

svc:/system/extended-accounting:flow
  Flow accounting
svc:/system/extended-accounting:process
  Process accounting
svc:/system/extended-accounting:task
  Task accounting
Enabling extended accounting by using `acctadm(1M)` causes the corresponding service instance to be enabled if not currently enabled, so that the extended accounting configuration will be restored at the next boot. Similarly, if the configuration results in accounting being disabled for a service, the service instance will be disabled. The instances are enabled or disabled by `acctadm` as needed.

To permanently activate extended accounting for a resource, run:

```
# acctadm -e resource_list
```

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

### Records

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

### Commands Used With Extended Accounting

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

For information on commands that are associated with tasks and projects, see “Example Commands and Command Options” on page 56. For information on IPQoS flow accounting, see `ipqosconf(1M)`.
Perl Interface to libexacct

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

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

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

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

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

The following table summarizes each of the modules.

<table>
<thead>
<tr>
<th>Module (should not contain spaces)</th>
<th>Description</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun::Solaris::Project</td>
<td>This module provides functions to access the project manipulation functions getprojid(2), endprojent(3PROJECT), fgetprojent(3PROJECT), getdefaultproj(3PROJECT), getprojbyid(3PROJECT), getprojbyname(3PROJECT), getprojent(3PROJECT), getprojidbyname(3PROJECT), inproj(3PROJECT), project_walk(3PROJECT), setproject(3PROJECT), and setprojent(3PROJECT).</td>
<td>Project(3PERL)</td>
</tr>
<tr>
<td>Sun::Solaris::Task</td>
<td>This module provides functions to access the task manipulation functions gettaskid(2) and settaskid(2).</td>
<td>Task(3PERL)</td>
</tr>
<tr>
<td>Module (should not contain spaces)</td>
<td>Description</td>
<td>For More Information</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Sun::Solaris::Exacct</td>
<td>This module is the top-level <code>exacct</code> module. This module provides functions to access the <code>exacct</code>-related system calls <code>getacct(2)</code>, <code>putacct(2)</code>, and <code>wraacct(2)</code>. This module also provides functions to access the <code>libexacct(3LIB)</code> library function <code>ea_error(3EXACCT)</code>. Constants for all of the <code>exacct EO_</code>, <code>EW_</code>, <code>EXR_</code>, <code>P_</code>, and <code>TASK_*</code> macros are also provided in this module.</td>
<td><code>exacct(3PERL)</code></td>
</tr>
<tr>
<td>Sun::Solaris::Exacct::Catalog</td>
<td>This module provides object-oriented methods to access the bitfields in an <code>exacct</code> catalog tag. This module also provides access to the constants for the <code>EXC_*</code>, <code>EXD_*</code>, and <code>EXD_*</code> macros.</td>
<td><code>exacct::Catalog(3PERL)</code></td>
</tr>
<tr>
<td>Sun::Solaris::Exacct::File</td>
<td>This module provides object-oriented methods to access the <code>libexacct</code> accounting file functions <code>ea_open(3EXACCT)</code>, <code>ea_close(3EXACCT)</code>, <code>ea_get_creator(3EXACCT)</code>, <code>ea_get_hostname(3EXACCT)</code>, <code>ea_next_object(3EXACCT)</code>, <code>ea_previous_object(3EXACCT)</code>, and <code>ea_write_object(3EXACCT)</code>.</td>
<td><code>exacct::File(3PERL)</code></td>
</tr>
<tr>
<td>Sun::Solaris::Exacct::Object</td>
<td>This module provides object-oriented methods to access an individual <code>exacct</code> accounting file object. An <code>exacct</code> object is represented as an opaque reference blessed into the appropriate <code>Sun::Solaris::Exacct::Object</code> subclass. This module is further subdivided into the object types <code>Item</code> and <code>Group</code>. At this level, there are methods to access the <code>ea_match_object_catalog(3EXACCT)</code> and <code>ea_attach_to_object(3EXACCT)</code> functions.</td>
<td><code>exacct::Object(3PERL)</code></td>
</tr>
<tr>
<td>Sun::Solaris::Exacct::Object::Item</td>
<td>This module provides object-oriented methods to access an individual <code>exacct</code> accounting file <code>Item</code>. Objects of this type inherit from <code>Sun::Solaris::Exacct::Object</code>.</td>
<td><code>exacct::Object::Item(3PERL)</code></td>
</tr>
<tr>
<td>Sun::Solaris::Exacct::Object::Group</td>
<td>This module provides object-oriented methods to access an individual <code>exacct</code> accounting file <code>Group</code>. Objects of this type inherit from <code>Sun::Solaris::Exacct::Object</code>. These objects provide access to the <code>ea_attach_to_group(3EXACCT)</code> function. The <code>Items</code> contained within the <code>Group</code> are presented as a Perl array.</td>
<td><code>exacct::Object::Group(3PERL)</code></td>
</tr>
<tr>
<td>Module (should not contain spaces)</td>
<td>Description</td>
<td>For More Information</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Sun::Solaris::Kstat</td>
<td>This module provides a Perl tied hash interface to the kstat facility. A usage example for this module can be found in /bin/kstat, which is written in Perl.</td>
<td>Kstat(3PERL)</td>
</tr>
</tbody>
</table>

For examples that show how to use the modules described in the previous table, see “Using the Perl Interface to libexecacct” on page 79.
Administering Extended Accounting (Tasks)

This chapter describes how to administer the extended accounting subsystem.

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

Administering the Extended Accounting Facility (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate the extended accounting facility.</td>
<td>Use extended accounting to monitor resource consumption by each project running on your system. You can use the extended accounting subsystem to capture historical data for tasks, processes, and flows.</td>
<td>&quot;How to Activate Extended Accounting for Flows, Processes, Tasks, and Network Componentss” on page 76</td>
</tr>
<tr>
<td>Display extended accounting status.</td>
<td>Determine the status of the extended accounting facility.</td>
<td>&quot;How to Display Extended Accounting Status” on page 77</td>
</tr>
<tr>
<td>View available accounting resources.</td>
<td>View the accounting resources available on your system.</td>
<td>&quot;How to View Available Accounting Resources” on page 77</td>
</tr>
<tr>
<td>Deactivate the flow, process, task, and net accounting instances.</td>
<td>Turn off the extended accounting functionality.</td>
<td>&quot;How to Deactivate Process, Task, Flow, and Network Management Accounting” on page 78</td>
</tr>
<tr>
<td>Use the Perl interface to the extended accounting facility.</td>
<td>Use the Perl interface to develop customized reporting and extraction scripts.</td>
<td>&quot;Using the Perl Interface to Libexacct” on page 79</td>
</tr>
</tbody>
</table>
Using Extended Accounting Functionality

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

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

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

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

Note – The System Administrator role includes the Process Management profile. For information on how to create the role and assign the role to a user, see Managing RBAC (Task Map) in System Administration Guide: Security Services.

1. Become superuser or assume an equivalent role.
   
   ```
   # acctadm -e extended -f /var/adm/exacct/proc process
   ```
3. Activate extended accounting for tasks.
   
   ```
   # acctadm -e extended,mstate -f /var/adm/exacct/task task
   ```
   
   ```
   # acctadm -e extended -f /var/adm/exacct/flow flow
   ```
5. Activate extended accounting for network.
   
   ```
   # acctadm -e extended -f /var/adm/exacct/net net
   ```

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

See Also   See acctadm(1M) for more information.
How to Display Extended Accounting Status

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

```
machine% acctadm
```

Task accounting: active
Task accounting file: /var/adm/exacct/task
Tracked task resources: extended
Untracked task resources: none

Process accounting: active
Process accounting file: /var/adm/exacct/proc
Tracked process resources: extended
Untracked process resources: host

Flow accounting: active
Flow accounting file: /var/adm/exacct/flow
Tracked flow resources: extended
Untracked flow resources: none

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

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

How to View Available Accounting Resources

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

```
machine% acctadm -r
```

```
process:
exended pid,uid,gid,cpu,time,command/tty,projid,taskid,ancpid,wait-status,zone,flag
memory,mstate          displays as one line
basic     pid,uid,gid,cpu,time,command/tty,flag

task:
exended taskid,projid,cpu,time:host,mstate,anctaskid,zone
basic     taskid,projid,cpu,time

flow:
exended
basic     saddr,daddr,sport,dport,proto,dsfield,nbytes,npkts,action,ctime,lseen,projid,uid
```

Chapter 5 • Administering Extended Accounting (Tasks) 77
How to Deactivate Process, Task, Flow, and Network Management Accounting

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

1. Become superuser or assume an equivalent role.

2. Turn off process accounting.
   ```
   # acctadm -x process
   ```

3. Turn off task accounting.
   ```
   # acctadm -x task
   ```

4. Turn off flow accounting.
   ```
   # acctadm -x flow
   ```

5. Turn off network management accounting.
   ```
   # acctadm -x net
   ```

6. Verify that task accounting, process accounting, flow and network accounting have been turned off.
   ```
   # acctadm
   Task accounting: inactive
   Task accounting file: none
   Tracked task resources: none
   Untracked task resources: extended
   Process accounting: inactive
   Process accounting file: none
   Tracked process resources: none
   Untracked process resources: extended
   Flow accounting: inactive
   Flow accounting file: none
   Tracked flow resources: none
   Untracked flow resources: extended
   ```
Using the Perl Interface to libexacct

How to Recursively Print the Contents of an exact Object

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

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

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

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

# Retrieve the value of the item. If the item contains
# in turn a nested exact object (i.e., an item or
# group), then the value method will return a reference
# to the appropriate sort of perl object
# (Exacct::Object::Item or Exactt::Object::Group).
# We could of course figure out that the item contained
# a nested item or group by examining the catalog tag in
# @cat and looking for a type of EXT_EXACCT_OBJECT or
# EXT_GROUP.
# my $val = $obj->value();
if (ref($val)) {
    # If it is a nested object, recurse to dump it.
    dump_object($val, $indent);
} else {
    # Otherwise it is just a 'plain' value, so
    # display it.
    printf("%s Value = %s
", $istr, $val);
}

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

    printf("%sGROUP\n%s Catalog = %s|%s|%s\n", $istr, $istr, @cat);
    $indent++;
    # 'foreach' provides a list context.
    foreach my $val ($obj->value()) {
        dump_object($val, $indent);
    }
    printf("%sENDGROUP\n", $istr);
}

Using the Perl Interface to libexacct
How to Create a New Group Record and Write It to a File

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

#!/usr/bin/perl

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

# Prototype list of catalog tags and values.
my @items = (
    [ &EXT_STRING | &EXC_DEFAULT | &EXD_CREATOR => "me" ],
    [ &EXT_UINT32 | &EXC_DEFAULT | &EXD_PROC_PID => $$ ],
    [ &EXT_UINT32 | &EXC_DEFAULT | &EXD_PROC_UID => $< ],
    [ &EXT_UINT32 | &EXC_DEFAULT | &EXD_PROC_GID => $( ],
    [ &EXT_STRING | &EXC_DEFAULT | &EXD_PROC_COMMAND => "/bin/rec" ],
);

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

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

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

# Open the exacct file, write the record & close.
my $f = ea_new_file('/tmp/exacct', &O_RDWR | &O_CREAT | &O_TRUNC)
|| die("create /tmp/exacct failed:", ea_error_str(), 

How to Print the Contents of an exacct File

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

#!/usr/bin/perl

use strict;
use warnings;
use Sun::Solaris::Exacct qw(:EXACCT_ALL);
die("Usage is dumpexact <exact file>\n") unless (@ARGV == 1);

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

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

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

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

Here is example output produced by running Sun::Solaris::Exact::Object->dump() on the file created in “How to Create a New Group Record and Write It to a File” on page 81.

Creator: root
Hostname: localhost

GROUP
	Catalog = EXT_GROUP|EXC_DEFAULT|EXD_NONE
	ITEM
		Catalog = EXT_STRING|EXC_DEFAULT|EXD_CREATOR
		Value = me
	ITEM
		Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_PID
		Value = 845523
	ITEM
		Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_UID
		Value = 37845
	ITEM
		Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_GID
		Value = 10
	ITEM
		Catalog = EXT_STRING|EXC_DEFAULT|EXD_PROC_COMMAND
		Value = /bin/rec
ENDGROUP
After you determine the resource consumption of workloads on your system as described in Chapter 4, “Extended Accounting (Overview),” you can place boundaries on resource usage. Boundaries prevent workloads from over-consuming resources. The resource controls facility is the constraint mechanism that is used for this purpose.

This chapter covers the following topics.

- “Resource Controls Concepts” on page 83
- “Configuring Resource Controls and Attributes” on page 85
- “Applying Resource Controls” on page 96
- “Temporarily Updating Resource Control Values on a Running System” on page 96
- “Commands Used With Resource Controls” on page 97

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

**Resource Controls Concepts**

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

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

For a list of the standard resource controls that are available in this release, see “Available Resource Controls” on page 86. See “Resource Type Properties” on page 234 for information on available zone-wide resource controls.
Resource Limits and Resource Controls

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

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

Interprocess Communication and Resource Controls

Processes can communicate with each other by using one of several types of interprocess communication (IPC). IPC allows information transfer or synchronization to occur between processes. Prior to the Solaris 10 release, IPC tunable parameters were set by adding an entry to the /etc/system file. The resource controls facility now provides resource controls that define the behavior of the kernel’s IPC facilities. These resource controls replace the /etc/system tunables.

Obsolete parameters might be included in the /etc/system file on this Solaris system. If so, the parameters are used to initialize the default resource control values as in previous Solaris releases. However, using the obsolete parameters is not recommended.

To observe which IPC objects are contributing to a project’s usage, use the ipcs command with the -J option. See “How to Use ipcs” on page 107 to view an example display. For more information about the ipcs command, see ipcs(1).

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

Resource Control Constraint Mechanisms

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

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

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

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

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

Configuring Resource Controls and Attributes

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

devolution:101:Developers:::task.max-lwps=(privileged,10,deny);
     process.max-cpu-time=(basic,3600,signal=TERM),(priv,3600,signal=KILL)

typed as one line

Note – On systems that have zones enabled, zone-wide resource controls are specified in the zone configuration using a slightly different format. See “Zone Configuration Data” on page 229 for more information.
The `rctladm` command allows you to make runtime interrogations of and modifications to the resource controls facility, with *global scope*. The `prctl` command allows you to make runtime interrogations of and modifications to the resource controls facility, with *local scope*.

For more information, see “Global and Local Actions on Resource Control Values” on page 92, `rctladm(1M)` and `prctl(1)`.

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

### Available Resource Controls

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

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

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

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

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

<table>
<thead>
<tr>
<th>Control Name</th>
<th>Description</th>
<th>Default Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>project.cpu-cap</code></td>
<td>Absolute limit on the amount of CPU resources that can be consumed by a project. A value of 100 means 100% of one CPU as the <code>project.cpu-cap</code> setting. A value of 125 is 125%, because 100% corresponds to one full CPU on the system when using CPU caps.</td>
<td>Quantity (number of CPUs)</td>
</tr>
<tr>
<td><code>project.cpu-shares</code></td>
<td>Number of CPU shares granted to this project for use with the fair share scheduler (see FSS(7)).</td>
<td>Quantity (shares)</td>
</tr>
<tr>
<td>Control Name</td>
<td>Description</td>
<td>Default Unit</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>project.max-crypto-memory</td>
<td>Total amount of kernel memory that can be used by libpkcs11 for hardware crypto acceleration. Allocations for kernel buffers and session-related structures are charged against this resource control.</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td>project.max-locked-memory</td>
<td>Total amount of physical locked memory allowed.</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td></td>
<td>If priv_proc_lock_memory is assigned to a user, consider setting this resource control as well to prevent that user from locking all memory.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note that this resource control replaced project.max-device-locked-memory, which has been removed. This release control will be removed in a future release.</td>
<td></td>
</tr>
<tr>
<td>project.max-msg-ids</td>
<td>Maximum number of message queue IDs allowed for this project.</td>
<td>Quantity</td>
</tr>
<tr>
<td></td>
<td>(message queue IDs)</td>
<td></td>
</tr>
<tr>
<td>project.max-port-ids</td>
<td>Maximum allowable number of event ports.</td>
<td>Quantity</td>
</tr>
<tr>
<td></td>
<td>(number of event ports)</td>
<td></td>
</tr>
<tr>
<td>project.max-sem-ids</td>
<td>Maximum number of semaphore IDs allowed for this project.</td>
<td>Quantity</td>
</tr>
<tr>
<td></td>
<td>(semaphore IDs)</td>
<td></td>
</tr>
<tr>
<td>project.max-shm-ids</td>
<td>Maximum number of shared memory IDs allowed for this project.</td>
<td>Quantity</td>
</tr>
<tr>
<td></td>
<td>(shared memory IDs)</td>
<td></td>
</tr>
<tr>
<td>project.max-shm-memory</td>
<td>Total amount of System V shared memory allowed for this project.</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td>project.max-lwps</td>
<td>Maximum number of LWP$s$ simultaneously available to this project.</td>
<td>Quantity</td>
</tr>
<tr>
<td></td>
<td>(LWP$s$)</td>
<td></td>
</tr>
<tr>
<td>project.max-tasks</td>
<td>Maximum number of tasks allowable in this project.</td>
<td>Quantity</td>
</tr>
<tr>
<td></td>
<td>(number of tasks)</td>
<td></td>
</tr>
<tr>
<td>project.max-contracts</td>
<td>Maximum number of contracts allowed in this project.</td>
<td>Quantity</td>
</tr>
<tr>
<td></td>
<td>(contracts)</td>
<td></td>
</tr>
<tr>
<td>task.max-cpu-time</td>
<td>Maximum CPU time that is available to this task’s processes.</td>
<td>Time (seconds)</td>
</tr>
<tr>
<td>task.max-lwps</td>
<td>Maximum number of LWP$s$ simultaneously available to this task’s processes.</td>
<td>Quantity</td>
</tr>
<tr>
<td></td>
<td>(LWP$s$)</td>
<td></td>
</tr>
</tbody>
</table>
### Table 6–1: Standard Project, Task, and Process Resource Controls (Continued)

<table>
<thead>
<tr>
<th>Control Name</th>
<th>Description</th>
<th>Default Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>process.max-cpu-time</td>
<td>Maximum CPU time that is available to this process.</td>
<td>Time (seconds)</td>
</tr>
<tr>
<td>process.max-file-descriptor</td>
<td>Maximum file descriptor index available to this process.</td>
<td>Index (maximum file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>descriptor)</td>
</tr>
<tr>
<td>process.max-file-size</td>
<td>Maximum file offset available for writing by this process.</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td>process.max-core-size</td>
<td>Maximum size of a core file created by this process.</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td>process.max-data-size</td>
<td>Maximum heap memory available to this process.</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td>process.max-stack-size</td>
<td>Maximum stack memory segment available to this process.</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td>process.max-address-space</td>
<td>Maximum amount of address space, as summed over segment sizes, that is</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td></td>
<td>available to this process.</td>
<td></td>
</tr>
<tr>
<td>process.max-port-events</td>
<td>Maximum allowable number of events per event port.</td>
<td>Quantity (number of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>events)</td>
</tr>
<tr>
<td>process.max-sem-nsems</td>
<td>Maximum number of semaphores allowed per semaphore set.</td>
<td>Quantity (semaphores</td>
</tr>
<tr>
<td></td>
<td></td>
<td>per set)</td>
</tr>
<tr>
<td>process.max-sem-ops</td>
<td>Maximum number of semaphore operations allowed per semop call (value</td>
<td>Quantity (number of</td>
</tr>
<tr>
<td></td>
<td>copied from the resource control at semget() time).</td>
<td>operations)</td>
</tr>
<tr>
<td>process.max-msg-qbytes</td>
<td>Maximum number of bytes of messages on a message queue (value copied from</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td></td>
<td>the resource control at msgget() time).</td>
<td></td>
</tr>
<tr>
<td>process.max-msg-messages</td>
<td>Maximum number of messages on a message queue (value copied from the</td>
<td>Quantity (number of</td>
</tr>
<tr>
<td></td>
<td>resource control at msgget() time).</td>
<td>messages)</td>
</tr>
</tbody>
</table>

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

Zone-wide resource controls limit the total resource usage of all process entities within a zone. Zone-wide resource controls can also be set using global property names as described in “Setting Zone-Wide Resource Controls” on page 222 and "How to Configure the Zone” on page 249.

TABLE 6-2  Zones Resource Controls

<table>
<thead>
<tr>
<th>Control Name</th>
<th>Description</th>
<th>Default Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone.cpu-cap</td>
<td>Absolute limit on the amount of CPU resources that can be consumed by a non-global zone. A value of 100 means 100% of one CPU as the project.cpu-cap setting. A value of 125 is 125%, because 100% corresponds to one full CPU on the system when using CPU caps.</td>
<td>Quantity (number of CPUs)</td>
</tr>
<tr>
<td>zone.cpu-shares</td>
<td>Number of fair share scheduler (FSS) CPU shares for this zone</td>
<td>Quantity (shares)</td>
</tr>
<tr>
<td>zone.max-locked-memory</td>
<td>Total amount of physical locked memory available to a zone. When priv.proc.lock_memory is assigned to a zone, consider setting this resource control as well to prevent that zone from locking all memory.</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td>zone.max-lwps</td>
<td>Maximum number of LWP’s simultaneously available to this zone</td>
<td>Quantity (LWP’s)</td>
</tr>
<tr>
<td>zone.max-msg-ids</td>
<td>Maximum number of message queue IDs allowed for this zone</td>
<td>Quantity (message queue IDs)</td>
</tr>
<tr>
<td>zone.max-sem-ids</td>
<td>Maximum number of semaphore IDs allowed for this zone</td>
<td>Quantity (semaphore IDs)</td>
</tr>
<tr>
<td>zone.max-shm-ids</td>
<td>Maximum number of shared memory IDs allowed for this zone</td>
<td>Quantity (shared memory IDs)</td>
</tr>
<tr>
<td>zone.max-shm-memory</td>
<td>Total amount of System V shared memory allowed for this zone</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td>zone.max-swap</td>
<td>Total amount of swap that can be consumed by user process address space mappings and tmpfs mounts for this zone.</td>
<td>Size (bytes)</td>
</tr>
</tbody>
</table>
For information on configuring zone-wide resource controls, see “Resource Type Properties” on page 234 and “How to Configure the Zone” on page 249. To use zone-wide resource controls in \texttt{lx} branded zones, see “How to Configure, Verify, and Commit the \texttt{lx} Branded Zone” on page 441.

Note that it is possible to apply a zone-wide resource control to the global zone. See “Using the Fair Share Scheduler on a Solaris System With Zones Installed” on page 397 for additional information.

### Units Support

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

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

The following global flags are available:

<table>
<thead>
<tr>
<th>Global Flag</th>
<th>Resource Control Type String</th>
<th>Modifier</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCTL_GLOBAL_BYTES</td>
<td>bytes</td>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KB</td>
<td>$2^{10}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MB</td>
<td>$2^{20}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GB</td>
<td>$2^{30}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TB</td>
<td>$2^{40}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PB</td>
<td>$2^{50}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EB</td>
<td>$2^{60}$</td>
</tr>
<tr>
<td>RCTL_GLOBAL_SECONDS</td>
<td>seconds</td>
<td>s</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ks</td>
<td>$10^3$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ms</td>
<td>$10^6$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gs</td>
<td>$10^9$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ts</td>
<td>$10^{12}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ps</td>
<td>$10^{15}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Es</td>
<td>$10^{18}$</td>
</tr>
</tbody>
</table>
Scaled values can be used with resource controls. The following example shows a scaled threshold value:

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

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

## Resource Control Values and Privilege Levels

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

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

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

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

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

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

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

Global Actions on Resource Control Values

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

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

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

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

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

Local Actions on Resource Control Values

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

none

No action is taken on resource requests for an amount that is greater than the threshold. This action is useful for monitoring resource usage without affecting the progress of applications. You can also enable a global message that displays when the resource control is exceeded, although the process exceeding the threshold is not affected.

deny

You can deny resource requests for an amount that is greater than the threshold. For example, a task.max-lwps resource control with action deny causes a fork system call to fail if the new process would exceed the control value. See the fork(2) man page.
You can enable a global signal message action when the resource control is exceeded. A signal is sent to the process when the threshold value is exceeded. Additional signals are not sent if the process consumes additional resources. Available signals are listed in Table 6–3.

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

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

**TABLE 6–3  ** Signals Available to Resource Control Values

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

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

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

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

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

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

The global flags indicate the following.

- **lowerable**  
  Superuser privileges are not required to lower the privileged values for this control.

- **no-deny**  
  Even when threshold values are exceeded, access to the resource is never denied.

- **cpu-time**  
  SIGXCPU is available to be sent when threshold values of this resource are reached.

- **seconds**  
  The time value for the resource control.

- **no-basic**  
  Resource control values with the privilege type basic cannot be set. Only privileged resource control values are allowed.

- **no-signal**  
  A local signal action cannot be set on resource control values.

- **no-syslog**  
  The global syslog message action may not be set for this resource control.

- **deny**  
  Always deny request for resource when threshold values are exceeded.

- **count**  
  A count (integer) value for the resource control.

- **bytes**  
  Unit of size for the resource control.

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

**Resource Control Enforcement**

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

**Global Monitoring of Resource Control Events**

Often, the resource consumption of processes is unknown. To get more information, try using the global resource control actions that are available with the `rctladm` command. Use `rctladm` to establish a `syslog` action on a resource control. Then, if any entity managed by that resource

---

$ prctl -n process.max-cpu-time $

<table>
<thead>
<tr>
<th>NAME</th>
<th>PRIVILEGE</th>
<th>VALUE</th>
<th>FLAG</th>
<th>ACTION</th>
<th>RECIPIENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>process.max-cpu-time</td>
<td>privileged</td>
<td>18.4Es inf</td>
<td>signal=XCPU</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>system</td>
<td>18.4Es inf</td>
<td>none</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 6–1  Process Collectives, Container Relationships, and Their Resource Control Sets**
control encounters a threshold value, a system message is logged at the configured logging level. See Chapter 7, "Administering Resource Controls (Tasks)," and the rctladm(1M) man page for more information.

**Applying Resource Controls**

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

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

**Temporarily Updating Resource Control Values on a Running System**

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

**Updating Logging Status**

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

**Updating Resource Controls**

You can view and temporarily alter resource control values and actions on a per-process, per-task, or per-project basis by using the prctl command. A project, task, or process ID is given as input, and the command operates on the resource control at the level where the control is defined.
Any modifications to values and actions take effect immediately. However, these modifications apply to the current process, task, or project only. The changes are not recorded in the project database. If the system is restarted, the modifications are lost. Permanent changes to resource controls must be made in the project database.

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

### Commands Used With Resource Controls

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

<table>
<thead>
<tr>
<th>Command Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipcs(1)</td>
<td>Allows you to observe which IPC objects are contributing to a project's usage</td>
</tr>
<tr>
<td>prctl(1)</td>
<td>Allows you to make runtime interrogations of and modifications to the resource controls facility, with local scope</td>
</tr>
<tr>
<td>rctldm(1M)</td>
<td>Allows you to make runtime interrogations of and modifications to the resource controls facility, with global scope</td>
</tr>
</tbody>
</table>

The resource Controls(5) man page describes resource controls available through the project database, including units and scaling factors.
This chapter describes how to administer the resource controls facility.

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

## Administering Resource Controls (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set resource controls.</td>
<td>Set resource controls for a project in the /etc/project file.</td>
<td>“Setting Resource Controls” on page 100</td>
</tr>
<tr>
<td>Get or revise the resource control values for active processes, tasks, or projects, with local scope.</td>
<td>Make runtime interrogations of and modifications to the resource controls associated with an active process, task, or project on the system.</td>
<td>“Using the prctl Command” on page 102</td>
</tr>
<tr>
<td>On a running system, view or update the global state of resource controls.</td>
<td>View the global logging state of each resource control on a system-wide basis. Also set up the level of syslog logging when controls are exceeded.</td>
<td>“Using rctladm” on page 106</td>
</tr>
<tr>
<td>Report status of active interprocess communication (IPC) facilities.</td>
<td>Display information about active interprocess communication (IPC) facilities. Observe which IPC objects are contributing to a project’s usage.</td>
<td>“Using ipcs” on page 107</td>
</tr>
</tbody>
</table>
**Setting Resource Controls**

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

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

1. **Become superuser or assume an equivalent role.**

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

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

3. **View** the entry in the `/etc/project` **file** by using one of the following methods:
   - **Type:**
     ```
     # projects -l
     system
     projid : 0
     comment: **
     users : (none)
     groups : (none)
     attribs:
     .
     .
     .
     x-files
     projid : 100
     comment: **
     users : (none)
     groups : (none)
     attribs: task.max-lwps=(privileged,3,deny)
     ```
Sample Session

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

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

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

# csh /* creates second LWP */
# csh /* creates third LWP */
# csh /* cannot create more LWPs */
Vfork failed
```

### How to Set Multiple Controls on a Project

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

1. Become superuser or assume an equivalent role.
2 Use the `projmod` command with the `-s` and `-K` options to set resource controls on project `x-files`:

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

The following controls are set:

- A basic control with no action on the maximum LWPs per task.
- A privileged deny control on the maximum LWPs per task. This control causes any LWP creation that exceeds the maximum to fail, as shown in the previous example "How to Set the Maximum Number of LWPs for Each Task in a Project" on page 100.
- A limit on the maximum file descriptors per process at the basic level, which forces the failure of any open call that exceeds the maximum.

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

```bash
# projects -l
```

```
x-files
projid: 100
comment: **
users: (none)
groups: (none)
attrs: process.max-file-descriptor=(basic,128,deny)
task.max-lwps=(basic,10,none),(privileged,500,deny)
```

```bash
# cat etc/project
```

```
x-files:100:::process.max-file-descriptor=(basic,128,deny);
task.max-lwps=(basic,10,none),(privileged,500,deny)
```

---

**Using the `prctl` Command**

Use the `prctl` command to make runtime interrogations of and modifications to the resource controls associated with an active process, task, or project on the system. See the `prctl(1)` man page for more information.
### How to Use the `prctl` Command to Display Default Resource Control Values

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

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

```bash
# prctl $$
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>PRIVILEGE</th>
<th>VALUE</th>
<th>FLAG</th>
<th>ACTION</th>
<th>RECIPIENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>process.max-port-events</td>
<td>privileged</td>
<td>65.5K</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>2.15G</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.crypto-buffer-limit</td>
<td>system</td>
<td>16.0EB</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>18.4E</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.add-crypto-sessions</td>
<td>privileged</td>
<td>100</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>18.4E</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.min-crypto-sessions</td>
<td>privileged</td>
<td>20</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>18.4E</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-msg-messages</td>
<td>privileged</td>
<td>8.19K</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>4.29G</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-msg-qbytes</td>
<td>privileged</td>
<td>64.0KB</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>16.0EB</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-sem-ops</td>
<td>privileged</td>
<td>512</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>2.15G</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-sem-nsems</td>
<td>privileged</td>
<td>512</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>32.8K</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-address-space</td>
<td>privileged</td>
<td>16.0EB</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>16.0EB</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-file-descriptor</td>
<td>privileged</td>
<td>65.5K</td>
<td>-</td>
<td>deny</td>
<td>100337</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>2.15G</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-core-size</td>
<td>privileged</td>
<td>8.00EB</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>8.00EB</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
</tbody>
</table>
Using the `prctl` Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Privileged</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td>8.00MB</td>
<td>100337</td>
</tr>
<tr>
<td>privileged</td>
<td>8.00EB</td>
<td>-</td>
</tr>
<tr>
<td>process.max-data-size</td>
<td>16.0EB</td>
<td>max</td>
</tr>
<tr>
<td>privileged</td>
<td>16.0EB</td>
<td>max</td>
</tr>
<tr>
<td>system</td>
<td>16.0EB</td>
<td>max</td>
</tr>
<tr>
<td>process.max-file-size</td>
<td>8.00EB</td>
<td>max</td>
</tr>
<tr>
<td>privileged</td>
<td>8.00EB</td>
<td>max</td>
</tr>
<tr>
<td>system</td>
<td>8.00EB</td>
<td>max</td>
</tr>
<tr>
<td>process.max-cpu-time</td>
<td>18.4Es</td>
<td>inf</td>
</tr>
<tr>
<td>privileged</td>
<td>18.4Es</td>
<td>inf</td>
</tr>
<tr>
<td>system</td>
<td>18.4Es</td>
<td>none</td>
</tr>
<tr>
<td>task.max-cpu-time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>18.4Es</td>
<td>none</td>
</tr>
<tr>
<td>task.max-lwps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>2.15G</td>
<td>max</td>
</tr>
<tr>
<td>project.max-contracts</td>
<td>10.0K</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td>10.0K</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td>2.15G</td>
<td>max</td>
</tr>
<tr>
<td>project.max-device-locked-memory</td>
<td>499MB</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td>499MB</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td>16.0EB</td>
<td>max</td>
</tr>
<tr>
<td>project.max-port-ids</td>
<td>8.19K</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td>8.19K</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td>65.5K</td>
<td>max</td>
</tr>
<tr>
<td>project.max-shm-memory</td>
<td>1.95GB</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td>1.95GB</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td>16.0EB</td>
<td>max</td>
</tr>
<tr>
<td>project.max-shm-ids</td>
<td>128</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td>128</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td>16.8M</td>
<td>max</td>
</tr>
<tr>
<td>project.max-msg-ids</td>
<td>128</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td>128</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td>16.8M</td>
<td>max</td>
</tr>
<tr>
<td>project.max-sem-ids</td>
<td>128</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td>128</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td>16.8M</td>
<td>max</td>
</tr>
<tr>
<td>project.max-tasks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>2.15G</td>
<td>max</td>
</tr>
<tr>
<td>project.max-lwps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>2.15G</td>
<td>max</td>
</tr>
<tr>
<td>project.cpu-shares</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td>65.5K</td>
<td>max</td>
</tr>
<tr>
<td>zone.max-lwps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>2.15G</td>
<td>max</td>
</tr>
<tr>
<td>zone.cpu-shares</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td>65.5K</td>
<td>max</td>
</tr>
</tbody>
</table>
How to Use the prctl Command to Display Information for a Given Resource Control

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

```
# prctl -n process.max-file-descriptor $$
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>PRIVILEGE</th>
<th>VALUE</th>
<th>FLAG</th>
<th>ACTION</th>
<th>RECIPIENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>process.max-file-descriptor</td>
<td>basic</td>
<td>256</td>
<td>-</td>
<td>deny</td>
<td>110453</td>
</tr>
<tr>
<td></td>
<td>privileged</td>
<td>65.5K</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>2.15G</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
</tbody>
</table>

How to Use prctl to Temporarily Change a Value

This example procedure uses the prctl command to temporarily add a new privileged value to deny the use of more than three LWPs per project for the x-files project. The result is comparable to the result in “How to Set the Maximum Number of LWPs for Each Task in a Project” on page 100.

1. Become superuser or assume an equivalent role.

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

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

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

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

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

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

5. Verify the result.

   ```
   # prctl -n project.max-lwps -i project x-files
   ```
**How to Use `prctl` to Lower a Resource Control Value**

1. Become superuser or assume an equivalent role.

2. Use the `prctl` command with the `-r` option to change the lowest value of the `process.max-file-descriptor` resource control.
   
   ```bash
   # prctl -n process.max-file-descriptor -r -v 128 $$
   ```

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

1. Become superuser or assume an equivalent role.

2. Display the value of `project.cpu-shares` in the project `group.staff`.
   
   ```bash
   # prctl -n project.cpu-shares -i project group.staff
   project: 2: group.staff
   NAME        PRIVILEGE  VALUE  FLAG ACTION  RECIPIENT
   project.cpu-shares
     privileged    1    -    none       -
     system     65.5K  max    none
   ```

3. Replace the current `project.cpu-shares` value 1 with the value 10.
   
   ```bash
   # prctl -n project.cpu-shares -v 10 -r -i project group.staff
   ```

4. Display the value of `project.cpu-shares` in the project `group.staff`.
   
   ```bash
   # prctl -n project.cpu-shares -i project group.staff
   project: 2: group.staff
   NAME        PRIVILEGE  VALUE  FLAG ACTION  RECIPIENT
   project.cpu-shares
     privileged   10    -    none       -
     system     65.5K  max    none
   ```

**Using `rctladm`**

**How to Use `rctladm`**

Use the `rctladm` command to make runtime interrogations of and modifications to the global state of the resource controls facility. See the `rctladm(1M)` man page for more information.
For example, you can use `rctladm` with the `-e` option to enable the global syslog attribute of a resource control. When the control is exceeded, notification is logged at the specified syslog level. To enable the global syslog attribute of `process.max-file-descriptor`, type the following:

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

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

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

### Using `ipcs`

#### How to Use `ipcs`

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

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

```
# ipcs -J
```

```
IPC status from <running system> as of Wed Mar 26 18:53:15 PDT 2003
T ID KEY MODE OWNER GROUP PROJECT
Message Queues:

Shared Memory:
```

| m | 3600 | 0 | -rw-rw-rw- | uname | staff | x-files |
| m | 201  | 0 | -rw-rw-rw- | uname | staff | x-files |
| m | 1802 | 0 | -rw-rw-rw- | uname | staff | x-files |
| m | 503  | 0 | -rw-rw-rw- | uname | staff | x-files |
| m | 304  | 0 | -rw-rw-rw- | uname | staff | x-files |
```
Capacity Warnings

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

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

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

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

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

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

   # prctl -n task.max-lwps -v 40 -t privileged -d all 'pgrep httpd'

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

   # rctladm -e syslog task.max-lwps

3. Observe whether the workload trips the resource control.

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

   Jan  8 10:15:15 testmachine unix: [ID 859581 kernnotice]
   NOTICE: privileged rctl task.max-lwps exceeded by task 19
The analysis of workload data can indicate that a particular workload or group of workloads is monopolizing CPU resources. If these workloads are not violating resource constraints on CPU usage, you can modify the allocation policy for CPU time on the system. The fair share scheduling class described in this chapter enables you to allocate CPU time based on shares instead of the priority scheme of the timesharing (TS) scheduling class.

This chapter covers the following topics.

- “Introduction to the Scheduler” on page 109
- “CPU Share Definition” on page 110
- “CPU Shares and Process State” on page 111
- “CPU Share Versus Utilization” on page 111
- “CPU Share Examples” on page 111
- “FSS Configuration” on page 113
- “FSS and Processor Sets” on page 115
- “Combining FSS With Other Scheduling Classes” on page 117
- “Setting the Scheduling Class for the System” on page 118
- “Scheduling Class on a System with Zones Installed” on page 118
- “Commands Used With FSS” on page 118

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

**Introduction to the Scheduler**

A fundamental job of the operating system is to arbitrate which processes get access to the system’s resources. The process scheduler, which is also called the dispatcher, is the portion of the kernel that controls allocation of the CPU to processes. The scheduler supports the concept of scheduling classes. Each class defines a scheduling policy that is used to schedule processes within the class. The default scheduler in the Solaris Operating System, the TS scheduler, tries to
To give every process relatively equal access to the available CPUs, you might want to specify that certain processes be given more resources than others. You can use the *fair share scheduler* (FSS) to control the allocation of available CPU resources among workloads, based on their importance. This importance is expressed by the number of *shares* of CPU resources that you assign to each workload.

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

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

**Note** – If you are using the `project.cpu-shares` resource control on a Solaris system with zones installed, see “Zone Configuration Data” on page 229, “Resource Controls Used in Non-Global Zones” on page 366, and “Using the Fair Share Scheduler on a Solaris System With Zones Installed” on page 397.

---

**CPU Share Definition**

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

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

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

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

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

CPU Share Versus Utilization

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

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

CPU Share Examples

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

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

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

The next three examples illustrate how shares work in different configurations. These examples show that shares are only mathematically accurate for representing the usage if demand meets or exceeds available resources.
Example 1: Two CPU-Bound Processes in Each Project

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

Example 2: No Competition Between Projects

If $A$ and $B$ have only one CPU-bound process each, and $S_A = 1$ and $S_B = 100$, then the total number of shares is 101. Each project cannot use more than one CPU because each project has only one running process. Because no competition exists between projects for CPU resources in this configuration, projects $A$ and $B$ are each allocated 50 percent of all CPU resources. In this configuration, CPU share values are irrelevant. The projects' allocations would be the same (50/50), even if both projects were assigned zero shares.
Example 3: One Project Unable to Run

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

FSS Configuration

Projects and Users

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

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

For more information on the project(4) database and name services, see “project Database” on page 47.
CPU Shares Configuration

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

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

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

```
x-files:100:::project.cpu-shares={privileged,5,none}
```

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

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

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

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

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

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

As stated previously, processes that belong to projects with zero shares are always given zero system priority. Projects with one or more shares are running with priorities one and higher. Thus, projects with zero shares are only scheduled when CPU resources are available that are not requested by a nonzero share project.
The maximum number of shares that can be assigned to one project is 65535.

**FSS and Processor Sets**

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

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

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

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

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

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

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

**FSS and Processor Sets Examples**

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

Project A is running only on processor set 1. Project B is running on processor sets 1 and 2. Project C is running on processor sets 1, 2, and 3. Assume that each project has enough processes to utilize all available CPU power. Thus, there is always competition for CPU resources on each processor set.
The total system-wide project CPU allocations on such a system are shown in the following table.

<table>
<thead>
<tr>
<th>Project</th>
<th>Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project A</td>
<td>4% = (1/6 X 2/8)_{pset1}</td>
</tr>
<tr>
<td>Project B</td>
<td>28% = (2/6 X 2/8)<em>{pset1} + (2/5 X 4/8)</em>{pset2}</td>
</tr>
<tr>
<td>Project C</td>
<td>67% = (3/6 X 2/8)<em>{pset1} + (3/5 X 4/8)</em>{pset2} + (3/3 X 2/8)_{pset3}</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Project</th>
<th>Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project A</td>
<td>16.66% = (1/6)</td>
</tr>
<tr>
<td>Project B</td>
<td>33.33% = (2/6)</td>
</tr>
<tr>
<td>Project C</td>
<td>50% = (3/6)</td>
</tr>
</tbody>
</table>
Combining FSS With Other Scheduling Classes

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

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

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

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

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

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

```
$ ps -ef -o pset,class | grep -v CLS | sort | uniq
1 FSS
1 SYS
2 TS
2 RT
3 FX
```
Setting the Scheduling Class for the System

To set the default scheduling class for the system, see "How to Make FSS the Default Scheduler Class" on page 121, “Scheduling Class” on page 218, and dispadmin(1M). To move running processes into a different scheduling class, see “Configuring the FSS” on page 120 and priocntl(1).

Scheduling Class on a System with Zones Installed

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

The preferred way to use FSS in this case is to set FSS to be the system default scheduling class with the dispadmin command. All zones then benefit from getting a fair share of the system CPU resources. See “Scheduling Class” on page 218 for more information on scheduling class when zones are in use.

For information about moving running processes into a different scheduling class without changing the default scheduling class and rebooting, see Table 26–5 and the priocntl(1) man page.

Commands Used With FSS

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

<table>
<thead>
<tr>
<th>Command Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>priocntl(1)</td>
<td>Displays or sets scheduling parameters of specified processes, moves running processes into a different scheduling class.</td>
</tr>
<tr>
<td>ps(1)</td>
<td>Lists information about running processes, identifies in which scheduling classes processor sets are running.</td>
</tr>
<tr>
<td>dispadmin(1M)</td>
<td>Sets the default scheduler for the system. Also used to examine and tune the FSS scheduler’s time quantum value.</td>
</tr>
<tr>
<td>FSS(7)</td>
<td>Describes the fair share scheduler (FSS).</td>
</tr>
</tbody>
</table>
This chapter describes how to use the fair share scheduler (FSS).

For an overview of the FSS, see Chapter 8, "Fair Share Scheduler (Overview)." For information on scheduling class when zones are in use, see "Scheduling Class" on page 218.

### Administering the Fair Share Scheduler (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor CPU usage.</td>
<td>Monitor the CPU usage of projects, and projects in processor sets.</td>
<td>&quot;Monitoring the FSS&quot; on page 120</td>
</tr>
<tr>
<td>Set the default scheduler class.</td>
<td>Make a scheduler such as the FSS the default scheduler for the system.</td>
<td>&quot;How to Make FSS the Default Scheduler Class&quot; on page 121</td>
</tr>
<tr>
<td>Move running processes from one scheduler class to a different scheduling class, such as the FSS class.</td>
<td>Manually move processes from one scheduling class to another scheduling class without changing the default scheduling class and rebooting.</td>
<td>&quot;How to Manually Move Processes From the TS Class Into the FSS Class&quot; on page 121</td>
</tr>
<tr>
<td>Move all running processes from all scheduling classes to a different scheduling class, such as the FSS class.</td>
<td>Manually move processes in all scheduling classes to another scheduling class without changing the default scheduling class and rebooting.</td>
<td>&quot;How to Manually Move Processes From All User Classes Into the FSS Class&quot; on page 122</td>
</tr>
<tr>
<td>Move a project's processes into a different scheduling class, such as the FSS class.</td>
<td>Manually move a project’s processes from their current scheduling class to a different scheduling class.</td>
<td>&quot;How to Manually Move a Project’s Processes Into the FSS Class&quot; on page 122</td>
</tr>
</tbody>
</table>
Task Description For Information

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examine and tune FSS parameters.</td>
<td>Tune the scheduler's time quantum value. <em>Time quantum</em> is the amount of time that a thread is allowed to run before it must relinquish the processor.</td>
<td>“How to Tune Scheduler Parameters” on page 122</td>
</tr>
</tbody>
</table>

**Monitoring the FSS**

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

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

**How to Monitor System CPU Usage by Projects**

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

**How to Monitor CPU Usage by Projects in Processor Sets**

- To monitor the CPU usage of projects on a list of processor sets, type:
  
  `%
  prstat -J -c pset-list`

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

**Configuring the FSS**

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

Note that you can use `svcadm restart` to restart the scheduler service. See `svcadm(1M)` for more information.
How to Make FSS the Default Scheduler Class

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

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

1. Become superuser or assume an equivalent role.

2. Set the default scheduler for the system to be the FSS.
   
   ```
   # dispadmin -d FSS
   ```

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

3. Make this configuration take effect immediately, without rebooting.
   
   ```
   # priocntl -s -c FSS -i all
   ```

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

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

1. Become superuser or assume an equivalent role.

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

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

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

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

Note – All processes again run in the TS scheduling class after reboot.
How to Manually Move Processes From All User Classes Into the FSS Class

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

1. Become superuser or assume an equivalent role.

2. Move the init process (pid 1) into the FSS scheduling class.
   
   ```bash
   # priocntl -s -c FSS -i pid 1
   ```

3. Move all processes from their current scheduling classes into the FSS scheduling class.
   
   ```bash
   # priocntl -s -c FSS -i all
   ```

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

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

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

1. Become superuser or assume an equivalent role.

2. Move processes that run in project ID 10 to the FSS scheduling class.
   
   ```bash
   # priocntl -s -c FSS -i projid 10
   ```

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

How to Tune Scheduler Parameters

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

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

```
$ dispadmin -c FSS -g
```

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

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

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

```
$ dispadmin -c FSS -s file
```
Physical Memory Control Using the Resource Capping Daemon (Overview)

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

The following topics are covered in this chapter.

- “Introduction to the Resource Capping Daemon” on page 125
- “How Resource Capping Works” on page 126
- “Attribute to Limit Physical Memory Usage for Projects” on page 126
- “rcapd Configuration” on page 127
- “Monitoring Resource Utilization With rcapstat” on page 131
- “Commands Used With rcapd” on page 132

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

Introduction to the Resource Capping Daemon

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

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

Like the resource control, the resource cap can be defined by using attributes of project entries in the project database. However, while resource controls are synchronously enforced by the kernel, resource caps are asynchronously enforced at the user level by the resource capping daemon. With asynchronous enforcement, a small delay occurs as a result of the sampling interval used by the daemon.
For information about `rcapd`, see the `rcapd(1M)` man page. For information about projects and the project database, see Chapter 2, “Projects and Tasks (Overview),” and the `project(4)` man page. For information about resource controls, see Chapter 6, “Resource Controls (Overview).”

How Resource Capping Works

The daemon repeatedly samples the resource utilization of projects that have physical memory caps. The sampling interval used by the daemon is specified by the administrator. See “Determining Sample Intervals” on page 131 for additional information. When the system’s physical memory utilization exceeds the threshold for cap enforcement, and other conditions are met, the daemon takes action to reduce the resource consumption of projects with memory caps to levels at or below the caps.

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

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

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

Attribute to Limit Physical Memory Usage for Projects

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

```
rcap.max-rss
```

The total amount of physical memory, in bytes, that is available to processes in the project.

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

```
db:100::db,root::rcap.max-rss=10737418240
```
Note – The system might round the specified cap value to a page size.

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

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

rcapd **Configuration**

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

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

To configure the daemon, you must have superuser privileges or have the Process Management profile in your list of profiles. The System Administrator role includes the Process Management profile.

Configuration changes can be incorporated into `rcapd` according to the configuration interval (see "rcapd Operation Intervals" on page 130) or on demand by sending a `SIGHUP` (see the `kill`(1) man page).

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

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

**Using the Resource Capping Daemon on a System With Zones Installed**

You can control resident set size (RSS) usage of a zone by setting the `capped-memory` resource when you configure the zone. For more information, see "Physical Memory Control and the `capped-memory` Resource" on page 219. You can run `rcapd` in a zone, including the global zone, to enforce memory caps on projects in that zone.

You can set a temporary cap for the maximum amount of memory that can be consumed by a specified zone, until the next reboot. See "How to Specify a Temporary Resource Cap for a Zone" on page 137.
If you are using rcapd on a zone to regulate physical memory consumption by processes running in projects that have resource caps defined, you must configure the daemon in those zones.

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

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

**Memory Cap Enforcement Threshold**

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

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

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

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

To set the value for cap enforcement, see "How to Set the Memory Cap Enforcement Threshold" on page 135.

**Determining Cap Values**

If a project cap is set too low, there might not be enough memory for the workload to proceed effectively under normal conditions. The paging that occurs because the workload requires more memory has a negative effect on system performance.
Projects that have caps set too high can consume available physical memory before their caps are exceeded. In this case, physical memory is effectively managed by the kernel and not by rcapd.

In determining caps on projects, consider these factors.

**Impact on I/O system**

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

**Impact on CPU usage**

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

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

Another portion of the daemon's CPU time is spent enforcing caps when they are exceeded. The time spent is proportional to the amount of virtual memory involved. CPU time spent increases or decreases in response to corresponding changes in the total size of a workload's address space. This information is reported in the \texttt{vm} column of \texttt{rcapstat} output. See "Monitoring Resource Utilization With \texttt{rcapstat}" on page 131 and the \texttt{rcapstat(1)} man page for more information.

**Reporting on shared memory**

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

The estimate is usable with workloads such as databases, which utilize shared memory extensively. For database workloads, you can also sample a project's regular usage to determine a suitable initial cap value by using output from the \texttt{-J} or \texttt{-Z} options of the \texttt{prstat} command. For more...
rcapd Configuration

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

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

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

To tune intervals, see “How to Set Operation Intervals” on page 136.

**Determining rcapd Scan Intervals**

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

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

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

Monitoring Resource Utilization With `rcapstat`

Use `rcapstat` to monitor the resource utilization of capped projects. To view an example `rcapstat` report, see “Producing Reports With `rcapstat`” on page 138.

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

- `interval` Specifies the sampling interval in seconds. The default interval is 5 seconds.
- `count` Specifies the number of times that the statistics are repeated. By default, `rcapstat` reports statistics until a termination signal is received or until the `rcapd` process exits.

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

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

<table>
<thead>
<tr>
<th><code>rcapstat</code> Column Headings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The project ID of the capped project.</td>
</tr>
<tr>
<td>project</td>
<td>The project name.</td>
</tr>
<tr>
<td>nproc</td>
<td>The number of processes in the project.</td>
</tr>
<tr>
<td>vm</td>
<td>The total amount of virtual memory size used by processes in the project, including all mapped files and devices, in kilobytes (K), megabytes (M), or gigabytes (G).</td>
</tr>
</tbody>
</table>
### rcapstat Column Headings

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rss</td>
<td>The estimated amount of the total resident set size (RSS) of the processes in the project, in kilobytes (K), megabytes (M), or gigabytes (G), not accounting for pages that are shared.</td>
</tr>
<tr>
<td>cap</td>
<td>The RSS cap defined for the project. See &quot;Attribute to Limit Physical Memory Usage for Projects&quot; on page 126 or the rcapd(1M) manual page for information about how to specify memory caps.</td>
</tr>
<tr>
<td>at</td>
<td>The total amount of memory that rcapd attempted to page out since the last rcapstat sample.</td>
</tr>
<tr>
<td>avgat</td>
<td>The average amount of memory that rcapd attempted to page out during each sample cycle that occurred since the last rcapstat sample. The rate at which rcapd samples collection RSS can be set with rcapadm. See &quot;rcapd Operation Intervals&quot; on page 130.</td>
</tr>
<tr>
<td>pg</td>
<td>The total amount of memory that rcapd successfully paged out since the last rcapstat sample.</td>
</tr>
<tr>
<td>avgpg</td>
<td>An estimate of the average amount of memory that rcapd successfully paged out during each sample cycle that occurred since the last rcapstat sample. The rate at which rcapd samples process RSS sizes can be set with rcapadm. See &quot;rcapd Operation Intervals&quot; on page 130.</td>
</tr>
</tbody>
</table>

### Commands Used With rcapd

<table>
<thead>
<tr>
<th>Command Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rcapstat(1)</td>
<td>Monitors the resource utilization of capped projects.</td>
</tr>
<tr>
<td>rcapadm(1M)</td>
<td>Configures the resource capping daemon, displays the current status of the resource capping daemon if it has been configured, and enables or disables resource capping. Also used to set a temporary memory cap.</td>
</tr>
<tr>
<td>rcapd(1M)</td>
<td>The resource capping daemon.</td>
</tr>
</tbody>
</table>
Administering the Resource Capping Daemon (Tasks)

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

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

Setting the Resident Set Size Cap

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

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

1. **Become superuser, or assume a role that includes the Process Management profile.**
   The System Administrator role includes the Process Management profile. For information on how to create the role and assign the role to a user, see Managing RBAC (Task Map) in System Administration Guide: Security Services.

2. **Add this attribute to the /etc/project file:**
   rcap.max-rss=value

   **Example 11–1 RSS Project Cap**
   The following line in the /etc/project file sets an RSS cap of 10 Gbytes for a project named db.
   
   db:100::db,root::rcap.max-rss=10737418240
   
   Note that the system might round the specified cap value to a page size.
How to Use the projmod Command to Add an rcap.max-rss Attribute for a Project

1. Become superuser, or assume a role that includes the Process Management profile.
   The System Administrator role includes the Process Management profile. For information on how to create the role and assign the role to a user, see Managing RBAC (Task Map) in System Administration Guide: Security Services.

2. Set an rcap.max-rss attribute of 10 Gbytes in the /etc/project file, in this case for a project named db.
   
   # projmod -a -K rcap.max-rss=10GB db
   
   The /etc/project file then contains the line:
   
   db:100::db,root::rcap.max-rss=10737418240

Configuring and Using the Resource Capping Daemon (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the memory cap enforcement threshold.</td>
<td>Configure a cap that will be enforced when the physical memory available to processes is low.</td>
<td>&quot;How to Set the Memory Cap Enforcement Threshold&quot; on page 135</td>
</tr>
<tr>
<td>Set the operation interval.</td>
<td>The interval is applied to the periodic operations performed by the resource capping daemon.</td>
<td>&quot;How to Set Operation Intervals&quot; on page 136</td>
</tr>
<tr>
<td>Enable resource capping.</td>
<td>Activate resource capping on your system.</td>
<td>&quot;How to Enable Resource Capping&quot; on page 136</td>
</tr>
<tr>
<td>Disable resource capping.</td>
<td>Deactivate resource capping on your system.</td>
<td>&quot;How to Disable Resource Capping&quot; on page 137</td>
</tr>
<tr>
<td>Report cap and project information.</td>
<td>View example commands for producing reports.</td>
<td>&quot;Reporting Cap and Project Information&quot; on page 138</td>
</tr>
<tr>
<td>Monitor a project’s resident set size.</td>
<td>Produce a report on the resident set size of a project.</td>
<td>&quot;Monitoring the RSS of a Project&quot; on page 139</td>
</tr>
<tr>
<td>Determine a project’s working set size.</td>
<td>Produce a report on the working set size of a project.</td>
<td>&quot;Determining the Working Set Size of a Project&quot; on page 139</td>
</tr>
</tbody>
</table>
Administering the Resource Capping Daemon With \texttt{rcapadm}

This section contains procedures for configuring the resource capping daemon with \texttt{rcapadm}. See \texttt{"rcapd Configuration" on page 127} and the \texttt{rcapadm(1M)} man page for more information. Using the \texttt{rcapadm} to specify a temporary resource cap for a zone is also covered.

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

\begin{itemize}
\item \textbf{How to Set the Memory Cap Enforcement Threshold}
\end{itemize}

Caps can be configured so that they will not be enforced until the physical memory available to processes is low. See \texttt{"Memory Cap Enforcement Threshold" on page 128} for more information.

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

\begin{enumerate}
\item \textbf{Become superuser, or assume a role that includes the Process Management profile.}

The System Administrator role includes the Process Management profile. For information on how to create the role and assign the role to a user, see Managing RBAC (Task Map) in \textit{System Administration Guide: Security Services}.

\item \textbf{Use the \texttt{-c} option of \texttt{rcapadm} to set a different physical memory utilization value for memory cap enforcement.}

\begin{verbatim}
# rcapadm -c percent
\end{verbatim}

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

\textbf{See Also} To display the current physical memory utilization and the cap enforcement threshold, see \texttt{"Reporting Memory Utilization and the Memory Cap Enforcement Threshold" on page 140.}
How to Set Operation Intervals

“rcapd Operation Intervals” on page 130 contains information about the intervals for the periodic operations performed by rcapd. To set operation intervals using rcapadm, follow this procedure.

1  Become superuser, or assume a role that includes the Process Management profile.

The System Administrator role includes the Process Management profile. For information on how to create the role and assign the role to a user, see Managing RBAC (Task Map) in System Administration Guide: Security Services.

2  Use the -i option to set interval values.

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

Note – All interval values are specified in seconds.

How to Enable Resource Capping

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

1  Become superuser, or assume a role that includes the Process Management profile.

The System Administrator role includes the Process Management profile. For information on how to create the role and assign the role to a user, see Managing RBAC (Task Map) in System Administration Guide: Security Services.

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

   ■  Turn on resource capping using the svcadm command.

       # svcadm enable rcap

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

       # rcapadm -E

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

       # rcapadm -n -E
How to Disable Resource Capping

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

1. **Become superuser, or assume a role that includes the Process Management profile.**
   The System Administrator role includes the Process Management profile. For information on how to create the role and assign the role to a user, see Managing RBAC (Task Map) in System Administration Guide: Security Services.

2. **Disable the resource capping daemon in one of the following ways:**
   - **Turn off resource capping using the `svcadm` command.**
     
     ```
     # svcadm disable rcap
     ```
   - **To disable the resource capping daemon so that it will be stopped now and not be started when the system is booted, type:**
     
     ```
     # rcapadm -D
     ```
   - **To disable the resource capping daemon without stopping it, also specify the `-n` option:**
     
     ```
     # rcapadm -n -D
     ```

   **Tip – Disabling the Resource Capping Daemon Safely**

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

How to Specify a Temporary Resource Cap for a Zone

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

1. **Become superuser, or assume a role that includes the Process Management profile.**
   The System Administrator role includes the Process Management profile.

2. **Set a maximum memory value of 512 Mbytes for the zone** `my-zone`.
   
   ```
   # rcapadm -z testzone -m 512M
   ```
**Producing Reports With rcapstat**

Use rcapstat to report resource capping statistics. “Monitoring Resource Utilization With rcapstat” on page 131 explains how to use the rcapstat command to generate reports. That section also describes the column headings in the report. The rcapstat(1) man page also contains this information.

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

**Reporting Cap and Project Information**

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

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

```bash
user1machine% rcapstat 5 5
```

<table>
<thead>
<tr>
<th>id</th>
<th>project</th>
<th>nproc</th>
<th>vm</th>
<th>rss</th>
<th>cap</th>
<th>at</th>
<th>avgat</th>
<th>pg</th>
<th>avgpg</th>
</tr>
</thead>
<tbody>
<tr>
<td>112270</td>
<td>user1</td>
<td>24</td>
<td>123M</td>
<td>35M</td>
<td>50M</td>
<td>0K</td>
<td>3312K</td>
<td>0K</td>
<td></td>
</tr>
<tr>
<td>78194</td>
<td>user2</td>
<td>1</td>
<td>2368K</td>
<td>1856K</td>
<td>10M</td>
<td>0K</td>
<td>0K</td>
<td>0K</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>112270</td>
<td>user1</td>
<td>24</td>
<td>123M</td>
<td>35M</td>
<td>50M</td>
<td>0K</td>
<td>0K</td>
<td>0K</td>
<td></td>
</tr>
<tr>
<td>78194</td>
<td>user2</td>
<td>1</td>
<td>2368K</td>
<td>1928K</td>
<td>10M</td>
<td>0K</td>
<td>0K</td>
<td>0K</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>112270</td>
<td>user1</td>
<td>24</td>
<td>123M</td>
<td>35M</td>
<td>50M</td>
<td>0K</td>
<td>0K</td>
<td>0K</td>
<td></td>
</tr>
<tr>
<td>78194</td>
<td>user2</td>
<td>1</td>
<td>2368K</td>
<td>1928K</td>
<td>10M</td>
<td>0K</td>
<td>0K</td>
<td>0K</td>
<td></td>
</tr>
</tbody>
</table>

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

The subsequent reports show no significant activity.
Monitoring the RSS of a Project

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

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

```
user1machine% rcapstat 5 5
```

<table>
<thead>
<tr>
<th>id</th>
<th>project</th>
<th>nproc</th>
<th>vm</th>
<th>rss</th>
<th>cap</th>
<th>at</th>
<th>avgat</th>
<th>pg</th>
<th>avgpg</th>
</tr>
</thead>
<tbody>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6144M</td>
<td>690M</td>
<td>220M</td>
<td>5520K</td>
<td>131M</td>
<td>4912K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6144M</td>
<td>0M</td>
<td>131M</td>
<td>4912K</td>
<td>163K</td>
<td>163K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6171M</td>
<td>6144M</td>
<td>27M</td>
<td>147M</td>
<td>6048K</td>
<td>2016K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6146M</td>
<td>6144M</td>
<td>4872M</td>
<td>174M</td>
<td>4368K</td>
<td>1456K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6156M</td>
<td>6144M</td>
<td>12M</td>
<td>161M</td>
<td>3376K</td>
<td>1125K</td>
</tr>
</tbody>
</table>

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

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

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

---

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

Determining the Working Set Size of a Project

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

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

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

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

**Reporting Memory Utilization and the Memory Cap Enforcement Threshold**

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

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

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

<table>
<thead>
<tr>
<th>id</th>
<th>project</th>
<th>nproc</th>
<th>vm</th>
<th>rss</th>
<th>cap</th>
<th>at</th>
<th>avgat</th>
<th>pg</th>
<th>avgpg</th>
</tr>
</thead>
<tbody>
<tr>
<td>376565</td>
<td>rcap</td>
<td>0</td>
<td>0K</td>
<td>0K</td>
<td>10G</td>
<td>0K</td>
<td>0K</td>
<td>0K</td>
<td>0K</td>
</tr>
</tbody>
</table>

physical memory utilization: 55%  cap enforcement threshold: 0%
```
This chapter discusses the following features:

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

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

The following topics are covered in this chapter:

- “Introduction to Resource Pools” on page 144
- “Introduction to Dynamic Resource Pools” on page 145
- “About Enabling and Disabling Resource Pools and Dynamic Resource Pools” on page 145
- “Resource Pools Used in Zones” on page 145
- “When to Use Pools” on page 146
- “Resource Pools Framework” on page 147
- “Implementing Pools on a System” on page 149
- “project.pool Attribute” on page 149
- “SPARC: Dynamic Reconfiguration Operations and Resource Pools” on page 149
- “Creating Pools Configurations” on page 150
- “Directly Manipulating the Dynamic Configuration” on page 151
- “poold Overview” on page 151
- “Managing Dynamic Resource Pools” on page 151
- “Configuration Constraints and Objectives” on page 152
- “poold Features That Can Be Configured” on page 156
- “How Dynamic Resource Allocation Works” on page 159
- “Using poolstat to Monitor the Pools Facility and Resource Utilization” on page 162
- “Commands Used With the Resource Pools Facility” on page 163

For procedures using this functionality, see Chapter 13, “Creating and Administering Resource Pools (Tasks).”
**Introduction to Resource Pools**

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

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

---

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

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

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

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

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

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

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

About Enabling and Disabling Resource Pools and Dynamic Resource Pools

To enable and disable resource pools and dynamic resource pools, see “Enabling and Disabling the Pools Facility” on page 167.

Resource Pools Used in Zones

As an alternative to associating a zone with a configured resource pool on your system, you can use the zonecfg command to create a temporary pool that is in effect while the zone is running. See “dedicated-cpu Resource” on page 217 for more information.

On a system that has zones enabled, a non-global zone can be associated with one resource pool, although the pool need not be exclusively assigned to a particular zone. Moreover, you cannot bind individual processes in non-global zones to a different pool by using the poolbind command from the global zone. To associate a non-global zone with a pool, see “Configuring, Verifying, and Committing a Zone” on page 249.

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

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

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

For information about resource pool commands, see “Commands Used With the Resource Pools Facility” on page 163.
When to Use Pools

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

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

Create a real-time pool by using the RT scheduler and designated processor resources.

System utilization

Enforce system goals that you establish.

Use the automated pools daemon feature to identify available resources and then monitor workloads to detect when your specified objectives are no longer being satisfied. The daemon can take corrective action if possible, or the condition can be logged.

Resource Pools Framework

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

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

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

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

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

By default, the resource pools framework is not active. Resource pools must be enabled to create or modify the dynamic configuration. Static configuration files can be manipulated with the `poolcfg` or `libpool` commands even if the resource pools framework is disabled. Static configuration files cannot be created if the pools facility is not active. For more information on the configuration file, see “Creating Pools Configurations” on page 150.
The commands used with resource pools and the poold system daemon are described in the following man pages:

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

/etc/pooladm.conf Contents

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

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

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

Pools Properties

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

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

Datatree,pset.dbname=warehouse

For additional information about property types, see “poold Properties” on page 155.

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

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

- When the Solaris software boots, an init script checks to see if the /etc/pooladm.conf file exists. If this file is found and pools are enabled, then pooladm is invoked to make this configuration the active pools configuration. The system creates a dynamic configuration to reflect the organization that is requested in /etc/pooladm.conf, and the machine’s resources are partitioned accordingly.

- When the Solaris system is running, a pools configuration can either be activated if it is not already present, or modified by using the pooladm command. By default, the pooladm command operates on /etc/pooladm.conf. However, you can optionally specify an alternate location and file name, and use that file to update the pools configuration.

For information about enabling and disabling resource pools, see “Enabling and Disabling the Pools Facility” on page 167. The pools facility cannot be disabled when there are user-defined pools or resources in use.

To configure resource pools, you must have superuser privileges or have the Process Management profile in your list of profiles. The System Administrator role includes the Process Management profile.

The pooladm resource controller is started with the dynamic resource pools facility.

project.pool Attribute

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

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

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

SPARC: Dynamic Reconfiguration Operations and Resource Pools

Dynamic Reconfiguration (DR) enables you to reconfigure hardware while the system is running. A DR operation can increase, reduce, or have no effect on a given type of resource.
Because DR can affect available resource amounts, the pools facility must be included in these operations. When a DR operation is initiated, the pools framework acts to validate the configuration.

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

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

If you are using dynamic resource pools, note that it is possible for a partition to move out of pool control while the daemon is active. For more information, see “Identifying a Resource Shortage” on page 160.

### Creating Pools Configurations

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

- `system`
- `pool`
- `pset`
- `cpu`

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

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

- You can use the `pooladm` command with the `-s` option to discover the resources on the current system and place the results in a configuration file.
  
  This method is preferred. All active resources and components on the system that are capable of being manipulated by the pools facility are recorded. The resources include existing processor set configurations. You can then modify the configuration to rename the processor sets or to create additional pools if necessary.

- You can use the `poolcfg` command with the `-c` option and the `discover` or `create system name` subcommands to create a new pools configuration.
  
  These options are maintained for backward compatibility with previous releases.

Use `poolcfg` or `libpool` to modify the `/etc/pooladm.conf` file. Do not directly edit this file.
**Directly Manipulating the Dynamic Configuration**

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

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

For an example, see “Transferring Resources” on page 182.

If DRP is in use, note that the resource transfer might trigger action from `poold`. See “`poold Overview`” on page 151 for more information.

**`poold Overview`**

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

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

`poold` maintains a decision history that it can examine. The decision history is used to eliminate reconfigurations that historically did not show improvements.

Note that a reconfiguration can also be triggered asynchronously if the workload objectives are changed or if the resources available to the system are modified.

**Managing Dynamic Resource Pools**

The DRP service is managed by the service management facility (SMF) under the service identifier `svc:/system/pools/dynamic`.

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using the `svcadm` command. The service's status can be queried using the `svcs` command. See the `svcs(1)` and `svcadm(1M)` man pages for more information.
The SMF interface is the preferred method for controlling DRP, but for backward compatibility, the following methods can also be used.

- If dynamic resource allocation is not required, `poold` can be stopped with the SIGQUIT or the SIGTERM signal. Either of these signals causes `poold` to terminate gracefully.
- Although `poold` will automatically detect changes in the resource or pools configuration, you can also force a reconfiguration to occur by using the SIGHUP signal.

### Configuration Constraints and Objectives

When making changes to a configuration, `poold` acts on directions that you provide. You specify these directions as a series of constraints and objectives. `poold` uses your specifications to determine the relative value of different configuration possibilities in relation to the existing configuration. `poold` then changes the resource assignments of the current configuration to generate new candidate configurations.

### Configuration Constraints

Constraints affect the range of possible configurations by eliminating some of the potential changes that could be made to a configuration. The following constraints, which are specified in the `libpool` configuration, are available.

- The minimum and maximum CPU allocations
- Pinned components that are not available to be moved from a set
- The importance factor of the pool

See the `libpool(3LIB)` man page and “Pools Properties” on page 148 for more information about pools properties.

See “How to Set Configuration Constraints” on page 177 for usage instructions.

**pset.min Property and pset.max Property Constraints**

These two properties place limits on the number of processors that can be allocated to a processor set, both minimum and maximum. See Table 12–1 for more details about these properties.

Within these constraints, a resource partition’s resources are available to be allocated to other resource partitions in the same Solaris instance. Access to the resource is obtained by binding to a pool that is associated with the resource set. Binding is performed at login or manually by an administrator who has the PRIV_SYS_RES_CONFIG privilege.
**cpu.pinned Property Constraint**

The `cpu.pinned` property indicates that a particular CPU should not be moved by DRP from the processor set in which it is located. You can set this `libpool` property to maximize cache utilization for a particular application that is executing within a processor set.

See Table 12–1 for more details about this property.

**pool.importance Property Constraint**

The `pool.importance` property describes the relative importance of a pool as defined by the administrator.

### Configuration Objectives

Objectives are specified similarly to constraints. The full set of objectives is documented in Table 12–1.

There are two categories of objectives.

- **Workload dependent**
  
  A workload-dependent objective is an objective that will vary according to the nature of the workload running on the system. An example is the `utilization` objective. The utilization figure for a resource set will vary according to the nature of the workload that is active in the set.

- **Workload independent**
  
  A workload-independent objective is an objective that does not vary according to the nature of the workload running on the system. An example is the CPU `locality` objective. The evaluated measure of locality for a resource set does not vary with the nature of the workload that is active in the set.

You can define three types of objectives.

<table>
<thead>
<tr>
<th>Name</th>
<th>Valid Elements</th>
<th>Operators</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wt-load</code></td>
<td>system</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td><code>locality</code></td>
<td>pset</td>
<td>N/A</td>
<td>loose</td>
</tr>
<tr>
<td><code>utilization</code></td>
<td>pset</td>
<td>&lt; &gt; ~</td>
<td>0–100%</td>
</tr>
</tbody>
</table>

Objectives are stored in property strings in the `libpool` configuration. The property names are as follows:

- `system.poold.objectives`
Objectives have the following syntax:

- objectives = objective [; objective]*
- objective = [n:] keyword [op] [value]

All objectives take an optional importance prefix. The importance acts as a multiplier for the objective and thus increases the significance of its contribution to the objective function evaluation. The range is from 0 to INT64_MAX (9223372036854775807). If not specified, the default importance value is 1.

Some element types support more than one type of objective. An example is pset. You can specify multiple objective types for these elements. You can also specify multiple utilization objectives on a single pset element.

See "How to Define Configuration Objectives" on page 178 for usage examples.

**wt-load Objective**

The **wt-load** objective favors configurations that match resource allocations to resource utilizations. A resource set that uses more resources will be given more resources when this objective is active. **wt-load** means *weighted load*.

Use this objective when you are satisfied with the constraints you have established using the minimum and maximum properties, and you would like the daemon to manipulate resources freely within those constraints.

**The locality Objective**

The **locality** objective influences the impact that locality, as measured by locality group (**lgroup**) data, has upon the selected configuration. An alternate definition for locality is latency. An **lgroup** describes CPU and memory resources. The **lgroup** is used by the Solaris system to determine the distance between resources, using time as the measurement. For more information on the locality group abstraction, see "Locality Groups Overview" in Programming Interfaces Guide.

This objective can take one of the following three values:

- **tight** If set, configurations that maximize resource locality are favored.
- **loose** If set, configurations that minimize resource locality are favored.
- **none** If set, the favorableness of a configuration is not influenced by resource locality. This is the default value for the **locality** objective.

In general, the **locality** objective should be set to **tight**. However, to maximize memory bandwidth or to minimize the impact of DR operations on a resource set, you could set this objective to **loose** or keep it at the default setting of **none**.
utilization Objective

The utilization objective favors configurations that allocate resources to partitions that are not meeting the specified utilization objective.

This objective is specified by using operators and values. The operators are as follows:

- `<`: The “less than” operator indicates that the specified value represents a maximum target value.
- `>`: The “greater than” operator indicates that the specified value represents a minimum target value.
- `~`: The “about” operator indicates that the specified value is a target value about which some fluctuation is acceptable.

A pset can only have one utilization objective set for each type of operator.

- If the `~` operator is set, then the `<` and `>` operators cannot be set.
- If the `<` and `>` operators are set, then the `~` operator cannot be set. Note that the settings of the `<` operator and the `>` operator cannot contradict each other.

You can set both a `<` and `>` operator together to create a range. The values will be validated to make sure that they do not overlap.

Configuration Objectives Example

In the following example, poold is to assess these objectives for the pset:

- The utilization should be kept between 30 percent and 80 percent.
- The locality should be maximized for the processor set.
- The objectives should take the default importance of 1.

**EXAMPLE 12–1**  
* poold Objectives Example

```
pset.poold.objectives "utilization > 30; utilization < 80; locality tight"
```

See “How to Define Configuration Objectives” on page 178 for additional usage examples.

poold Properties

There are four categories of properties:

- Configuration
- Constraint
- Objective
### Objective Parameter

#### TABLE 12-1 Defined Property Names

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>system.poold.log-level</td>
<td>string</td>
<td>Configuration</td>
<td>Logging level</td>
</tr>
<tr>
<td>system.poold.log-location</td>
<td>string</td>
<td>Configuration</td>
<td>Logging location</td>
</tr>
<tr>
<td>system.poold.monitor-interval</td>
<td>uint64</td>
<td>Configuration</td>
<td>Monitoring sample interval</td>
</tr>
<tr>
<td>system.poold.history-file</td>
<td>string</td>
<td>Configuration</td>
<td>Decision history location</td>
</tr>
<tr>
<td>pset.max</td>
<td>uint64</td>
<td>Constraint</td>
<td>Maximum number of CPUs for this processor set</td>
</tr>
<tr>
<td>pset.min</td>
<td>uint64</td>
<td>Constraint</td>
<td>Minimum number of CPUs for this processor set</td>
</tr>
<tr>
<td>cpu.pinned</td>
<td>bool</td>
<td>Constraint</td>
<td>CPUs pinned to this processor set</td>
</tr>
<tr>
<td>system.poold.objectives</td>
<td>string</td>
<td>Objective</td>
<td>Formatted string following poold's objective expression syntax</td>
</tr>
<tr>
<td>pset.poold.objectives</td>
<td>string</td>
<td>Objective</td>
<td>Formatted string following poold's expression syntax</td>
</tr>
<tr>
<td>pool.importance</td>
<td>int64</td>
<td>Objective parameter</td>
<td>User-assigned importance</td>
</tr>
</tbody>
</table>

**poold Features That Can Be Configured**

You can configure these aspects of the daemon’s behavior.

- Monitoring interval
- Logging level
- Logging location

These options are specified in the pools configuration. You can also control the logging level from the command line by invoking `poold`.

---

**Features ThatCanBeConfigured**

You can configure these aspects of the daemon’s behavior.

- Monitoring interval
- Logging level
- Logging location

These options are specified in the pools configuration. You can also control the logging level from the command line by invoking `poold`.

---

**poold Features That Can Be Configured**

You can configure these aspects of the daemon’s behavior.

- Monitoring interval
- Logging level
- Logging location

These options are specified in the pools configuration. You can also control the logging level from the command line by invoking `poold`.
**poo1d Monitoring Interval**

Use the property name `system.poo1d.monitor-interval` to specify a value in milliseconds.

**poo1d Logging Information**

Three categories of information are provided through logging. These categories are identified in the logs:

- Configuration
- Monitoring
- Optimization

Use the property name `system.poo1d.log-level` to specify the logging parameter. If this property is not specified, the default logging level is NOTICE. The parameter levels are hierarchical. Setting a log level of DEBUG will cause poo1d to log all defined messages. The INFO level provides a useful balance of information for most administrators.

At the command line, you can use the `poo1d` command with the `-l` option and a parameter to specify the level of logging information generated.

The following parameters are available:

- ALERT
- CRIT
- ERR
- WARNING
- NOTICE
- INFO
- DEBUG

The parameter levels map directly onto their syslog equivalents. See “Logging Location” on page 159 for more information about using syslog.

For more information about how to configure poo1d logging, see “How to Set the poo1d Logging Level” on page 180.

**Configuration Information Logging**

The following types of messages can be generated:

<table>
<thead>
<tr>
<th>ALERT</th>
<th>Problems accessing the <code>libpool</code> configuration, or some other fundamental, unanticipated failure of the <code>libpool</code> facility. Causes the daemon to exit and requires immediate administrative attention.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRIT</td>
<td>Problems due to unanticipated failures. Causes the daemon to exit and requires immediate administrative attention.</td>
</tr>
</tbody>
</table>
### Monitoring Information Logging

The following types of messages can be generated:

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRIT</td>
<td>Problems due to unanticipated monitoring failures. Causes the daemon to exit and requires immediate administrative attention.</td>
</tr>
<tr>
<td>ERR</td>
<td>Problems due to unanticipated monitoring error. Could require administrative intervention to correct.</td>
</tr>
<tr>
<td>NOTICE</td>
<td>Messages about resource control region transitions.</td>
</tr>
<tr>
<td>INFO</td>
<td>Messages about resource utilization statistics.</td>
</tr>
<tr>
<td>DEBUG</td>
<td>Messages containing the detailed information that is needed when debugging monitoring processing. This information is not generally used by administrators.</td>
</tr>
</tbody>
</table>

### Optimization Information Logging

The following types of messages can be generated:

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WARNING</td>
<td>Messages could be displayed regarding problems making optimal decisions. Examples could include resource sets that are too narrowly constrained by their minimum and maximum values or by the number of pinned components. Messages could be displayed about problems performing an optimal reallocation due to unforeseen limitations. Examples could include removing the last processor from a processor set which contains a bound resource consumer.</td>
</tr>
<tr>
<td>NOTICE</td>
<td>Messages about usable configurations or configurations that will not be implemented due to overriding decision histories could be displayed.</td>
</tr>
<tr>
<td>INFO</td>
<td>Messages about alternate configurations considered could be displayed.</td>
</tr>
</tbody>
</table>
Messages containing the detailed information that is needed when debugging optimization processing. This information is not generally used by administrators.

Logging Location

The system.poold.log.location property is used to specify the location for poold logged output. You can specify a location of SYSLOG for poold output (see syslog(3C)).

If this property is not specified, the default location for poold logged output is 
/var/log/pool/poold.

When poold is invoked from the command line, this property is not used. Log entries are written to stderr on the invoking terminal.

Log Management With logadm

If poold is active, the logadm.conf file includes an entry to manage the default file 
/var/log/pool/poold. The entry is:

/var/log/pool/poold -N -s 512k

See the logadm(1M) and the logadm.conf(4) man pages.

How Dynamic Resource Allocation Works

This section explains the process and the factors that poold uses to dynamically allocate resources.

About Available Resources

Available resources are considered to be all of the resources that are available for use within the scope of the poold process. The scope of control is at most a single Solaris instance.

On a system that has zones enabled, the scope of an executing instance of poold is limited to the global zone.
**Determining Available Resources**

Resource pools encompass all of the system resources that are available for consumption by applications.

For a single executing Solaris instance, a resource of a single type, such as a CPU, must be allocated to a single partition. There can be one or more partitions for each type of resource. Each partition contains a unique set of resources.

For example, a machine with four CPUs and two processor sets can have the following setup:

- pset 0: 0 1
- pset 1: 2 3

where 0, 1, 2 and 3 after the colon represent CPU IDs. Note that the two processor sets account for all four CPUs.

The same machine cannot have the following setup:

- pset 0: 0 1
- pset 1: 1 2 3

It cannot have this setup because CPU 1 can appear in only one pset at a time.

Resources cannot be accessed from any partition other than the partition to which they belong.

To discover the available resources, `poold` interrogates the active pools' configuration to find partitions. All resources within all partitions are summed to determine the total amount of available resources for each type of resource that is controlled.

This quantity of resources is the basic figure that `poold` uses in its operations. However, there are constraints upon this figure that limit the flexibility that `poold` has to make allocations. For information about available constraints, see “Configuration Constraints” on page 152.

**Identifying a Resource Shortage**

The control scope for `poold` is defined as the set of available resources for which `poold` has primary responsibility for effective partitioning and management. However, other mechanisms that are allowed to manipulate resources within this control scope can still affect a configuration. If a partition should move out of control while `poold` is active, `poold` tries to restore control through the judicious manipulation of available resources. If `poold` cannot locate additional resources within its scope, then the daemon logs information about the resource shortage.
Determining Resource Utilization

pool typically spends the greatest amount of time observing the usage of the resources within its scope of control. This monitoring is performed to verify that workload-dependent objectives are being met.

For example, for processor sets, all measurements are made across all of the processors in a set. The resource utilization shows the proportion of time that the resource is in use over the sample interval. Resource utilization is displayed as a percentage from 0 to 100.

Identifying Control Violations

The directives described in “Configuration Constraints and Objectives” on page 152 are used to detect the approaching failure of a system to meet its objectives. These objectives are directly related to workload.

A partition that is not meeting user-configured objectives is a control violation. The two types of control violations are synchronous and asynchronous.

- A synchronous violation of an objective is detected by the daemon in the course of its workload monitoring.
- An asynchronous violation of an objective occurs independently of monitoring action by the daemon.

The following events cause asynchronous objective violations:

- Resources are added to or removed from a control scope.
- The control scope is reconfigured.
- The pool resource controller is restarted.

The contributions of objectives that are not related to workload are assumed to remain constant between evaluations of the objective function. Objectives that are not related to workload are only reassessed when a reevaluation is triggered through one of the asynchronous violations.

Determining Appropriate Remedial Action

When the resource controller determines that a resource consumer is short of resources, the initial response is that increasing the resources will improve performance.

Alternative configurations that meet the objectives specified in the configuration for the scope of control are examined and evaluated.
This process is refined over time as the results of shifting resources are monitored and each resource partition is evaluated for responsiveness. The decision history is consulted to eliminate reconfigurations that did not show improvements in attaining the objective function in the past. Other information, such as process names and quantities, are used to further evaluate the relevance of the historical data.

If the daemon cannot take corrective action, the condition is logged. For more information, see “poold Logging Information” on page 157.

**Using poolstat to Monitor the Pools Facility and Resource Utilization**

The poolstat utility is used to monitor resource utilization when pools are enabled on your system. This utility iteratively examines all of the active pools on a system and reports statistics based on the selected output mode. The poolstat statistics enable you to determine which resource partitions are heavily utilized. You can analyze these statistics to make decisions about resource reallocation when the system is under pressure for resources.

The poolstat utility includes options that can be used to examine specific pools and report resource set-specific statistics.

If zones are implemented on your system and you use poolstat in a non-global zone, information about the resources associated with the zone’s pool is displayed.

For more information about the poolstat utility, see the poolstat(1M) man page. For poolstat task and usage information, see "Using poolstat to Report Statistics for Pool-Related Resources” on page 186.

**poolstat Output**

In default output format, poolstat outputs a heading line and then displays a line for each pool. A pool line begins with the pool ID and the name of the pool, followed by a column of statistical data for the processor set attached to the pool. Resource sets attached to more than one pool are listed multiple times, once for each pool.

The column headings are as follows:

- **id** Pool ID.
- **pool** Pool name.
- **rid** Resource set ID.
- **rset** Resource set name.
**type**  Resource set type.

**min**  Minimum resource set size.

**max**  Maximum resource set size.

**size**  Current resource set size.

**used**  Measure of how much of the resource set is currently used.

This usage is calculated as the percentage of utilization of the resource set multiplied by the size of the resource set. If a resource set has been reconfigured during the last sampling interval, this value might be not reported. An unreported value appears as a hyphen (-).

**load**  Absolute representation of the load that is put on the resource set.

For more information about this property, see the `libpool(3LIB)` man page.

You can specify the following in `poolstat` output:

- The order of the columns
- The headings that appear

---

**Tuning poolstat Operation Intervals**

You can customize the operations performed by `poolstat`. You can set the sampling interval for the report and specify the number of times that statistics are repeated:

- **interval**  Tune the intervals for the periodic operations performed by `poolstat`. All intervals are specified in seconds.

- **count**  Specify the number of times that the statistics are repeated. By default, `poolstat` reports statistics only once.

If **interval** and **count** are not specified, statistics are reported once. If **interval** is specified and **count** is not specified, then statistics are reported indefinitely.

---

**Commands Used With the Resource Pools Facility**

The commands described in the following table provide the primary administrative interface to the pools facility. For information on using these commands on a system that has zones enabled, see "Resource Pools Used in Zones" on page 145.
## Commands Used With the Resource Pools Facility

<table>
<thead>
<tr>
<th>Man Page Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pooladm(1M)</code></td>
<td>Enables or disables the pools facility on your system. Activates a particular configuration or removes the current configuration and returns associated resources to their default status. If run without options, <code>pooladm</code> prints out the current dynamic pools configuration.</td>
</tr>
<tr>
<td><code>poolbind(1M)</code></td>
<td>Enables the manual binding of projects, tasks, and processes to a resource pool.</td>
</tr>
<tr>
<td><code>poolcfg(1M)</code></td>
<td>Provides configuration operations on pools and sets. Configurations created using this tool are instantiated on a target host by using <code>pooladm</code>. If run with the <code>info</code> subcommand argument to the <code>-c</code> option, <code>poolcfg</code> displays information about the static configuration at <code>/etc/pooladm.conf</code>. If a file name argument is added, this command displays information about the static configuration held in the named file. For example, <code>poolcfg -c info /tmp/newconfig</code> displays information about the static configuration contained in the file <code>/tmp/newconfig</code>.</td>
</tr>
<tr>
<td><code>poold(1M)</code></td>
<td>The pools system daemon. The daemon uses system targets and observable statistics to preserve the system performance goals specified by the administrator. If unable to take corrective action when goals are not being met, <code>poold</code> logs the condition.</td>
</tr>
<tr>
<td><code>poolstat(1M)</code></td>
<td>Displays statistics for pool-related resources. Simplifies performance analysis and provides information that supports system administrators in resource partitioning and repartitioning tasks. Options are provided for examining specified pools and reporting resource set-specific statistics.</td>
</tr>
</tbody>
</table>

A library API is provided by `libpool` (see the `libpool(3LIB)` man page). The library can be used by programs to manipulate pool configurations.
This chapter describes how to set up and administer resource pools on your system.

For background information about resource pools, see Chapter 12, “Resource Pools (Overview).”

### Administering Resource Pools (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable or disable resource pools.</td>
<td>Activate or disable resource pools on your system.</td>
<td>“Enabling and Disabling the Pools Facility” on page 167</td>
</tr>
<tr>
<td>Enable or disable dynamic resource pools.</td>
<td>Activate or disable dynamic resource pools facilities on your system.</td>
<td>“Enabling and Disabling the Pools Facility” on page 167</td>
</tr>
<tr>
<td>Create a static resource pools configuration.</td>
<td>Create a static configuration file that matches the current dynamic configuration. For more information, see “Resource Pools Framework” on page 147.</td>
<td>“How to Create a Static Configuration” on page 171</td>
</tr>
<tr>
<td>Modify a resource pools configuration.</td>
<td>Revise a pools configuration on your system, for example, by creating additional pools.</td>
<td>“How to Modify a Configuration” on page 173</td>
</tr>
<tr>
<td>Associate a resource pool with a scheduling class.</td>
<td>Associate a pool with a scheduling class so that all processes bound to the pool use the specified scheduler.</td>
<td>“How to Associate a Pool With a Scheduling Class” on page 175</td>
</tr>
</tbody>
</table>
**Task Description**

Set configuration constraints and define configuration objectives.

Specify objectives for `pool`d to consider when taking corrective action. For more information on configuration objectives, see "`pool`d Overview" on page 151.

Specify objectives for `pool`d to consider when taking corrective action. For more information on configuration objectives, see "`pool`d Overview" on page 151.

Set the logging level.

Specify the level of logging information that `pool`d generates.

"How to Set the `pool`d Logging Level" on page 180

Use a text file with the `pool`c fg command.

The `pool`c fg command can take input from a text file.

"How to Use Command Files With `pool`c fg" on page 181

Transfer resources in the kernel.

Transfer resources in the kernel. For example, transfer resources with specific IDs to a target set.

"Transferring Resources" on page 182

Activate a pools configuration.

Activate the configuration in the default configuration file.

"How to Activate a Pools Configuration" on page 183

Validate a pools configuration before you commit the configuration.

Validate a pools configuration to test what will happen when the validation occurs.

"How to Validate a Configuration Before Committing the Configuration" on page 183

Remove a pools configuration from your system.

All associated resources, such as processor sets, are returned to their default status.

"How to Remove a Pools Configuration" on page 183

Bind processes to a pool.

Manually associate a running process on your system with a resource pool.

"How to Bind Processes to a Pool" on page 184

Bind tasks or projects to a pool.

Associate tasks or projects with a resource pool.

"How to Bind Tasks or Projects to a Pool" on page 185

Bind new processes to a resource pool.

To automatically bind new processes in a project to a given pool, add an attribute to each entry in the project database.

"How to Set the project. `pool` Attribute for a Project" on page 185

Use project attributes to bind a process to a different pool.

Modify the pool binding for new processes that are started.

"How to Use project Attributes to Bind a Process to a Different Pool" on page 185

Use the `pool`stat utility to produce reports.

Produce multiple reports at specified intervals.

"Producing Multiple Reports at Specific Intervals" on page 187

Report resource set statistics.

Use the `pool`stat utility to report statistics for a psset resource set.

"Reporting Resource Set Statistics" on page 187

---

**Table of Tasks**

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set configuration constraints and</td>
<td>Specify objectives for <code>pool</code>d to consider when taking corrective action.</td>
<td>&quot;How to Set Configuration Constraints&quot; on page 177 and &quot;How to Define Configuration Objectives&quot; on page 178</td>
</tr>
<tr>
<td>define configuration objectives.</td>
<td>For more information on configuration objectives, see &quot;<code>pool</code>d Overview&quot; on page 151.</td>
<td></td>
</tr>
<tr>
<td>Set the logging level.</td>
<td>Specify the level of logging information that <code>pool</code>d generates.</td>
<td>&quot;How to Set the <code>pool</code>d Logging Level&quot; on page 180</td>
</tr>
<tr>
<td>Use a text file with the <code>pool</code>c fg</td>
<td>The <code>pool</code>c fg command can take input from a text file.</td>
<td>&quot;How to Use Command Files With <code>pool</code>c fg&quot; on page 181</td>
</tr>
<tr>
<td>command.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transfer resources in the kernel.</td>
<td>Transfer resources in the kernel. For example, transfer resources with</td>
<td>&quot;Transferring Resources&quot; on page 182</td>
</tr>
<tr>
<td></td>
<td>specific IDs to a target set.</td>
<td></td>
</tr>
<tr>
<td>Activate a pools configuration.</td>
<td>Activate the configuration in the default configuration file.</td>
<td>&quot;How to Activate a Pools Configuration&quot; on page 183</td>
</tr>
<tr>
<td>Validate a pools configuration before you</td>
<td>Validate a pools configuration to test what will happen when the validation</td>
<td>&quot;How to Validate a Configuration Before Committing the Configuration&quot; on page 183</td>
</tr>
<tr>
<td>commit the configuration.</td>
<td>occurs.</td>
<td></td>
</tr>
<tr>
<td>Remove a pools configuration from your</td>
<td>All associated resources, such as processor sets, are returned to their</td>
<td>&quot;How to Remove a Pools Configuration&quot; on page 183</td>
</tr>
<tr>
<td>system.</td>
<td>default status.</td>
<td></td>
</tr>
<tr>
<td>Bind processes to a pool.</td>
<td>Manually associate a running process on your system with a resource pool.</td>
<td>&quot;How to Bind Processes to a Pool&quot; on page 184</td>
</tr>
<tr>
<td>Bind tasks or projects to a pool.</td>
<td>Associate tasks or projects with a resource pool.</td>
<td>&quot;How to Bind Tasks or Projects to a Pool&quot; on page 185</td>
</tr>
<tr>
<td>Bind new processes to a resource pool.</td>
<td>To automatically bind new processes in a project to a given pool, add an</td>
<td>&quot;How to Set the project. <code>pool</code> Attribute for a Project&quot; on page 185</td>
</tr>
<tr>
<td></td>
<td>attribute to each entry in the project database.</td>
<td></td>
</tr>
<tr>
<td>Use project attributes to bind a process</td>
<td>Modify the pool binding for new processes that are started.</td>
<td>&quot;How to Use project Attributes to Bind a Process to a Different Pool&quot; on page 185</td>
</tr>
<tr>
<td>to a different pool.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use the <code>pool</code>stat utility to produce</td>
<td>Produce multiple reports at specified intervals.</td>
<td>&quot;Producing Multiple Reports at Specific Intervals&quot; on page 187</td>
</tr>
<tr>
<td>reports.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Enabling and Disabling the Pools Facility

You can enable and disable the resource pools and dynamic resource pools services on your system by using the `svcadm` command described in the `svcadm(1M)` man page.

You can also use the `pooladm` command described in the `pooladm(1M)` man page to perform the following tasks:

- Enable the pools facility so that pools can be manipulated
- Disable the pools facility so that pools cannot be manipulated

**Note** – When a system is upgraded, if the resource pools framework is enabled and an `/etc/pooladm.conf` file exists, the pools service is enabled and the configuration contained in the file is applied to the system.

**▼ How to Enable the Resource Pools Service Using `svcadm`**

1. Become superuser, or assume a role that includes the Process Management profile.
   The System Administrator role includes the Process Management profile. For more information about roles, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2. Enable the resource pools service.
   
   ```
   # svcadm enable system/pools:default
   ```

**▼ How to Disable the Resource Pools Service Using `svcadm`**

1. Become superuser, or assume a role that includes the Process Management profile.
   The System Administrator role includes the Process Management profile. For more information about roles, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2. Disable the resource pools service.
   
   ```
   # svcadm disable system/pools:default
   ```
How to Enable the Dynamic Resource Pools Service Using `svcadm`

1. Become superuser, or assume a role that includes the Service Management rights profile.

2. Enable the dynamic resource pools service.
   
   ```
   # svcadm enable system/pools/dynamic:default
   ```

**Example 13-1** Dependency of the Dynamic Resource Pools Service on the Resource Pools Service

This example shows that you must first enable resource pools if you want to run DRP.

There is a dependency between resource pools and dynamic resource pools. DRP is now a dependent service of resource pools. DRP can be independently enabled and disabled apart from resource pools.

The following display shows that both resource pools and dynamic resource pools are currently disabled:

```
# svcs *pool*
STATE STIME FMRI
disabled 10:32:26 svc:/system/pools/dynamic:default
disabled 10:32:26 svc:/system/pools:default
```

Enable dynamic resource pools:

```
# svcadm enable svc:/system/pools/dynamic:default
# svcs -a | grep pool
disabled 10:39:00 svc:/system/pools:default
offline 10:39:12 svc:/system/pools/dynamic:default
```

Note that the DRP service is still offline.

Use the `-x` option of the `svcs` command to determine why the DRP service is offline:

```
# svcs -x *pool*
svc:/system/pools:default (resource pools framework)
State: disabled since Wed 25 Jan 2006 10:39:00 AM GMT
Reason: Disabled by an administrator.
    See: http://sun.com/msg/SMF-8000-05
    See: libpool(3LIB)
    See: pooladm(1M)
    See: poolbind(1M)
    See: poolcfg(1M)
    See: poolstat(1M)
    See: /var/svc/log/system-pools:default.log
```
Impact: 1 dependent service is not running. (Use -v for list.)

svc:/system/pools/dynamic:default (dynamic resource pools)
State: offline since Wed 25 Jan 2006 10:39:12 AM GMT
Reason: Service svc:/system/pools:default is disabled.
  See: http://sun.com/msg/SMF-8000-GE
  See: poold(1M)
  See: /var/svc/log/system-pools-dynamic:default.log
Impact: This service is not running.

Enable the resource pools service so that the DRP service can run:

```
# svcadm enable svc:/system/pools:default
```

When the `svcs *pool*` command is used, the system displays:

```
# svcs *pool*
STATE   STIME    FMRI
online  10:40:27 svc:/system/pools:default
online  10:40:27 svc:/system/pools/dynamic:default
```

**Example 13-2** Effect on Dynamic Resource Pools When the Resource Pools Service Is Disabled

If both services are online and you disable the resource pools service:

```
# svcadm disable svc:/system/pools:default
```

When the `svcs *pool*` command is used, the system displays:

```
# svcs *pool*
STATE   STIME    FMRI
disabled 10:41:05 svc:/system/pools:default
online  10:40:27 svc:/system/pools/dynamic:default
```

But eventually, the DRP service moves to offline because the resource pools service has been disabled:

```
# svcs *pool*
STATE   STIME    FMRI
disabled 10:41:05 svc:/system/pools:default
offline 10:41:12 svc:/system/pools/dynamic:default
```

Determine why the DRP service is offline:
How to Disable the Dynamic Resource Pools Service Using `svcadm`

1. Become superuser, or assume a role that includes the Process Management profile.
   The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in `System Administration Guide: Basic Administration`

2. Disable the dynamic resource pools service.
   ```bash
   # svcadm disable system/pools/dynamic:default
   ```
### How to Enable Resource Pools Using `pooladm`

1. **Become superuser, or assume a role that includes the Process Management profile.**
   The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Enable the pools facility.**
   ```
   # pooladm -e
   ```

### How to Disable Resource Pools Using `pooladm`

1. **Become superuser, or assume a role that includes the Process Management profile.**
   The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Disable the pools facility.**
   ```
   # pooladm -d
   ```

### Configuring Pools

#### How to Create a Static Configuration

Use the `-s` option to `/usr/sbin/pooladm` to create a static configuration file that matches the current dynamic configuration. Unless a different file name is specified, the default location `/etc/pooladm.conf` is used.

Commit your configuration using the `pooladm` command with the `-c` option. Then, use the `pooladm` command with the `-s` option to update the static configuration to match the state of the dynamic configuration.

**Note** – The new functionality `pooladm -s` is preferred over the previous functionality `poolcfg -c discover` for creating a new configuration that matches the dynamic configuration.

**Before You Begin**

Enable pools on your system.
1. **Become superuser, or assume a role that includes the Process Management profile.**
   The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Update the static configuration file to match the current dynamic configuration.**
   ```
   # pooladm -s
   ```

3. **View the contents of the configuration file in readable form.**
   Note that the configuration contains default elements created by the system.
   ```
   # poolcfg -c info
   system tester
     string system.comment
     int system.version 1
     boolean system.bind-default true
     int system.poold.pid 177916

   pool pool_default
     int pool.sys_id 0
     boolean pool.active true
     boolean pool.default true
     int pool.importance 1
     string pool.comment
     pset pset_default

   pset pset_default
     int pset.sys_id -1
     boolean pset.default true
     uint pset.min 1
     uint pset.max 65536
     string pset.units population
     uint pset.load 10
     uint pset.size 4
     string pset.comment
     boolean testnullchanged true

   cpu
     int cpu.sys_id 3
     string cpu.comment
     string cpu.status on-line

   cpu
     int cpu.sys_id 2
     string cpu.comment
     string cpu.status on-line
   ```
How to Modify a Configuration

To enhance your configuration, create a processor set named pset_batch and a pool named pool_batch. Then join the pool and the processor set with an association.

Note that you must quote subcommand arguments that contain white space.

1 Become superuser, or assume a role that includes the Process Management profile.
The System Administrator role includes the Process Management profile. For more information about roles, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 Create processor set pset_batch.
   # poolcfg -c 'create pset pset_batch (uint pset.min = 2; uint pset.max = 10)'

3 Create pool pool_batch.
   # poolcfg -c 'create pool pool_batch'

4 Join the pool and the processor set with an association.
   # poolcfg -c 'associate pool pool_batch (pset pset_batch)'

5 Display the edited configuration.
   # poolcfg -c info
   system tester
       string system.comment kernel state
       int system.version 1
boolean system.bind-default true
int    system.poold.pid 177916

pool pool_default
    int    pool.sys_id 0
    boolean pool.active true
    boolean pool.default true
    int    pool.importance 1
    string pool.comment
    pset    pset_default

pset pset_default
    int    pset.sys_id -1
    boolean pset.default true
    int    pset.min 1
    uint   pset.max 65536
    string pset.units population
    uint   pset.load 10
    uint   pset.size 4
    string pset.comment
    boolean testnullchanged true

cpu
    int    cpu.sys_id 3
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 2
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 1
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 0
    string cpu.comment
    string cpu.status on-line

pool pool_batch
    boolean pool.default false
    boolean pool.active true
    int    pool.importance 1
    string pool.comment
    pset    pset_batch

Configuring Pools

pset pset_batch
  int pset.sys_id -2
  string pset.units population
  boolean pset.default true
  uint pset.max 10
  uint pset.min 2
  string pset.comment
  boolean pset.escapable false
  uint pset.load 0
  uint pset.size 0

cpu
  int cpu.sys_id 5
  string cpu.comment
  string cpu.status on-line

cpu
  int cpu.sys_id 4
  string cpu.comment
  string cpu.status on-line

6  Commit the configuration at /etc/pooladm.conf.
   # pooladm -c

7  (Optional) To copy the dynamic configuration to a static configuration file named /tmp/backup, type the following:
   # pooladm -s /tmp/backup

▼ How to Associate a Pool With a Scheduling Class

You can associate a pool with a scheduling class so that all processes bound to the pool use this scheduler. To do this, set the pool.scheduler property to the name of the scheduler. This example associates the pool pool_batch with the fair share scheduler (FSS).

1  Become superuser, or assume a role that includes the Process Management profile.
   The System Administrator role includes the Process Management profile. For more information about roles, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2  Modify pool pool_batch to be associated with the FSS.
   # poolcfg -c "modify pool pool_batch (string pool.scheduler="FSS")"
3 Display the edited configuration.

    # poolcfg -c info
    system tester
      string  system.comment
      int  system.version 1
      boolean  system.bind-default true
      int  system.poold.pid 177916

    pool pool_default
      int  pool.sys_id 0
      boolean  pool.active true
      boolean  pool.default true
      int  pool.importance 1
      string  pool.comment
      pset  pset_default

    pset pset_default
      int  pset.sys_id -1
      boolean  pset.default true
      uint  pset.min 1
      uint  pset.max 65536
      string  pset.units population
      uint  pset.load 10
      uint  pset.size 4
      string  pset.comment
      boolean  testnullchanged true

    cpu
      int  cpu.sys_id 3
      string  cpu.comment
      string  cpu.status on-line

    cpu
      int  cpu.sys_id 2
      string  cpu.comment
      string  cpu.status on-line

    cpu
      int  cpu.sys_id 1
      string  cpu.comment
      string  cpu.status on-line

    cpu
      int  cpu.sys_id 0
      string  cpu.comment
      string  cpu.status on-line

    pool pool_batch
4 Commit the configuration at /etc/pooladm.conf:
   # pooladm -c

5 (Optional) To copy the dynamic configuration to a static configuration file called /tmp/backup, type the following:
   # pooladm -s /tmp/backup

▼ How to Set Configuration Constraints

Constraints affect the range of possible configurations by eliminating some of the potential changes that could be made to a configuration. This procedure shows how to set the cpu.pinned property.

In the following examples, cpuid is an integer.
1 Become superuser, or assume a role that includes the Process Management profile.
The System Administrator role includes the Process Management profile. For more information about roles, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 Modify the cpu.pinned property in the static or dynamic configuration:
   - Modify the boot-time (static) configuration:
     # poolcfg -c 'modify cpu <cpuid> (boolean cpu.pinned = true)'
   - Modify the running (dynamic) configuration without modifying the boot-time configuration:
     # poolcfg -dc 'modify cpu <cpuid> (boolean cpu.pinned = true)'

▼ How to Define Configuration Objectives
You can specify objectives for poold to consider when taking corrective action.

In the following procedure, the wt-load objective is being set so that poold tries to match resource allocation to resource utilization. The locality objective is disabled to assist in achieving this configuration goal.

1 Become superuser, or assume a role that includes the Process Management profile.
The System Administrator role includes the Process Management profile. For more information about roles, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 Modify system tester to favor the wt-load objective.
   # poolcfg -c 'modify system tester (string system.poold.objectives="wt-load")'

3 Disable the locality objective for the default processor set.
   # poolcfg -c 'modify pset pset_default (string pset.poold.objectives="locality none")'  one line

4 Disable the locality objective for the pset_batch processor set.
   # poolcfg -c 'modify pset pset_batch (string pset.poold.objectives="locality none")'  one line

5 Display the edited configuration.
   # poolcfg -c info
   system tester
       string system.comment
       int     system.version 1
       boolean system.bind-default true
int  system.poold.pid 177916
string system.poold.objectives wt-load

pool pool_default
  int  pool.sys_id 0
  boolean pool.active true
  boolean pool.default true
  int  pool.importance 1
  string pool.comment
pset pset_default

  pset pset_default
  int  pset.sys_id -1
  boolean pset.default true
  uint pset.min 1
  uint pset.max 65536
  string pset.units population
  uint pset.load 10
  uint pset.size 4
  string pset.comment
  boolean testnullchanged true
  string pset.poold.objectives locality none

cpu
  int  cpu.sys_id 3
  string cpu.comment
  string cpu.status on-line

cpu
  int  cpu.sys_id 2
  string cpu.comment
  string cpu.status on-line

cpu
  int  cpu.sys_id 1
  string cpu.comment
  string cpu.status on-line

cpu
  int  cpu.sys_id 0
  string cpu.comment
  string cpu.status on-line

pool pool_batch
  boolean pool.default false
  boolean pool.active true
  int  pool.importance 1
  string pool.comment


```bash
string pool.schedul er FSS
pset batch

pset pset_batch
  int pset.sys_id -2
  string pset.units population
  boolean pset.default true
  uint pset.max 10
  uint pset.min 2
  string pset.comment
  boolean pset.escapable false
  uint pset.load 0
  uint pset.size 0
  string pset.poold.objectives locality none

cpu
  int cpu.sys_id 5
  string cpu.comment
  string cpu.status on-line

cpu
  int cpu.sys_id 4
  string cpu.comment
  string cpu.status on-line
```

6 Commit the configuration at `/etc/pooladm.conf`.
   # pooladm -c

7 (Optional) To copy the dynamic configuration to a static configuration file called `/tmp/backup`, type the following:
   # pooladm -s /tmp/backup

---

**How to Set the poold Logging Level**

To specify the level of logging information that poold generates, set the `system.poold.log-level` property in the poold configuration. The poold configuration is held in the `libpool` configuration. For information, see "poold Logging Information" on page 157 and the `poolcfg(1M)` and `libpool(3LIB)` man pages.

You can also use the poold command at the command line to specify the level of logging information that poold generates.
1 **Become superuser, or assume a role that includes the Process Management profile.**

   The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2 **Set the logging level by using the `poold` command with the `-l` option and a parameter, for example, `INFO`.**

   ```
   # /usr/lib/pool/poold -l INFO
   ```

   For information about available parameters, see “`poold` Logging Information” on page 157. The default logging level is `NOTICE`.

### How to Use Command Files With `poolcfg`

The `poolcfg` command with the `-f` option can take input from a text file that contains `poolcfg` subcommand arguments to the `-c` option. This method is appropriate when you want a set of operations to be performed. When processing multiple commands, the configuration is only updated if all of the commands succeed. For large or complex configurations, this technique can be more useful than per-subcommand invocations.

Note that in command files, the `#` character acts as a comment mark for the rest of the line.

1 **Create the input file** `poolcmds.txt`.

   ```
   $ cat > poolcmds.txt
   create system tester
   create pset pset_batch (uint pset.min = 2; uint pset.max = 10)
   create pool pool_batch
   associate pool pool_batch (pset pset_batch)
   ```

2 **Become superuser, or assume a role that includes the Process Management profile.**

   The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

3 **Execute the command:**

   ```
   # /usr/sbin/poolcfg -f poolcmds.txt
   ```
Transferring Resources

Use the `transfer` subcommand argument to the `-c` option of `poolcfg` with the `-d` option to transfer resources in the kernel. The `-d` option specifies that the command operate directly on the kernel and not take input from a file.

The following procedure moves two CPUs from processor set `pset1` to processor set `pset2` in the kernel.

▼ How to Move CPUs Between Processor Sets

1. **Become superuser, or assume a role that includes the Process Management profile.**
   The System Administrator role includes the Process Management profile. For more information about roles, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2. **Move two CPUs from `pset1` to `pset2`.**
   The `from` and `to` subclauses can be used in any order. Only one `to` and `from` subclause is supported per command.
   
   ```
   # poolcfg -dc 'transfer 2 from pset pset1 to pset2'
   ```

**Example 13–3** Alternative Method to Move CPUs Between Processor Sets

If specific known IDs of a resource type are to be transferred, an alternative syntax is provided. For example, the following command assigns two CPUs with IDs 0 and 2 to the `pset_large` processor set:

```
# poolcfg -dc "transfer to pset pset_large (cpu 0; cpu 2)"
```

More Information  Troubleshooting

If a transfer fails because there are not enough resources to match the request or because the specified IDs cannot be located, the system displays an error message.

Activating and Removing Pool Configurations

Use the `pooladm` command to make a particular pool configuration active or to remove the currently active pool configuration. See the `pooladm(1M)` man page for more information about this command.
How to Activate a Pools Configuration

To activate the configuration in the default configuration file, `/etc/pooladm.conf`, invoke `pooladm` with the `-c` option, “commit configuration.”

1. **Become superuser, or assume a role that includes the Process Management profile.**
   The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. **Commit the configuration at `/etc/pooladm.conf`.**
   ```
   # pooladm -c
   ```

3. **(Optional) Copy the dynamic configuration to a static configuration file, for example, `/tmp/backup`.**
   ```
   # pooladm -s /tmp/backup
   ```

How to Validate a Configuration Before Committing the Configuration

You can use the `-n` option with the `-c` option to test what will happen when the validation occurs. The configuration will not actually be committed.

The following command attempts to validate the configuration contained at `/home/admin/newconfig`. Any error conditions encountered are displayed, but the configuration itself is not modified.

1. **Become superuser, or assume a role that includes the Process Management profile.**
   The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. **Test the validity of the configuration before committing it.**
   ```
   # pooladm -n -c /home/admin/newconfig
   ```

How to Remove a Pools Configuration

To remove the current active configuration and return all associated resources, such as processor sets, to their default status, use the `-x` option for “remove configuration.”
Setting Pool Attributes and Binding to a Pool

1. Become superuser, or assume a role that includes the Process Management profile.
   The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. Remove the current active configuration.
   
   ```
   # pooladm -x
   ```
   
   The `-x` option to `pooladm` removes all user-defined elements from the dynamic configuration. All resources revert to their default states, and all pool bindings are replaced with a binding to the default pool.

More Information

Mixing Scheduling Classes Within a Processor Set

You can safely mix processes in the TS and IA classes in the same processor set. Mixing other scheduling classes within one processor set can lead to unpredictable results. If the use of `pooladm -x` results in mixed scheduling classes within one processor set, use the `priocntl` command to move running processes into a different scheduling class. See “How to Manually Move Processes From the TS Class Into the FSS Class” on page 121. Also see the `priocntl(1)` man page.

Setting Pool Attributes and Binding to a Pool

You can set a `project.pool` attribute to associate a resource pool with a project.

You can bind a running process to a pool in two ways:

- You can use the `poolbind` command described in `poolbind(1M)` command to bind a specific process to a named resource pool.
- You can use the `project.pool` attribute in the `project` database to identify the pool binding for a new login session or a task that is launched through the `newtask` command. See the `newtask(1)`, `projmod(1M)`, and `project(4)` man pages.

How to Bind Processes to a Pool

The following procedure uses `poolbind` with the `-p` option to manually bind a process (in this case, the current shell) to a pool named `ohare`.

1. Become superuser, or assume a role that includes the Process Management profile.
   The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.
2  Manually bind a process to a pool:
   # poolbind -p ohare $$

3  Verify the pool binding for the process by using poolbind with the -q option.
   $ poolbind -q $$
   155509 ohare
   The system displays the process ID and the pool binding.

▼  How to Bind Tasks or Projects to a Pool

To bind tasks or projects to a pool, use the poolbind command with the -i option. The following example binds all processes in the airmiles project to the laguardia pool.

1  Become superuser, or assume a role that includes the Process Management profile.
The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2  Bind all processes in the airmiles project to the laguardia pool.
   # poolbind -i project -p laguardia airmiles

▼  How to Set the project.pool Attribute for a Project

You can set the project.pool attribute to bind a project’s processes to a resource pool.

1  Become superuser, or assume a role that includes the Process Management profile.
The System Administrator role includes the Process Management profile. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2  Add a project.pool attribute to each entry in the project database.
   # projmod -a -K project.pool=poolname project

▼  How to Use project Attributes to Bind a Process to a Different Pool

Assume you have a configuration with two pools that are named studio and backstage. The /etc/project file has the following contents:
With this configuration, processes that are started by user paul are bound by default to the studio pool.

User paul can modify the pool binding for processes he starts. paul can use newtask to bind work to the backstage pool as well, by launching in the passes project.

1. **Launch a process in the passes project.**
   
   ```
   $ newtask -l -p passes
   ```

2. **Use the poolbind command with the -q option to verify the pool binding for the process. Also use a double dollar sign ($$) to pass the process number of the parent shell to the command.**
   
   ```
   $ poolbind -q $$
   ```

   The system displays the process ID and the pool binding.

---

**Using poolstat to Report Statistics for Pool-Related Resources**

The poolstat command is used to display statistics for pool-related resources. See "Using poolstat to Monitor the Pools Facility and Resource Utilization" on page 162 and the poolstat(1M) man page for more information.

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

**Displaying Default poolstat Output**

Typing poolstat without arguments outputs a header line and a line of information for each pool. The information line shows the pool ID, the name of the pool, and resource statistics for the processor set attached to the pool.

```bash
machine% poolstat

<table>
<thead>
<tr>
<th>pset</th>
<th>id</th>
<th>pool</th>
<th>size</th>
<th>used</th>
<th>load</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>pool_default</td>
<td>4</td>
<td>3.6</td>
<td>6.2</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>pool_sales</td>
<td>4</td>
<td>3.3</td>
<td>8.4</td>
</tr>
</tbody>
</table>
```
Producing Multiple Reports at Specific Intervals

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

```
machine% poolstat 5 3
```

<table>
<thead>
<tr>
<th>id</th>
<th>pool</th>
<th>size</th>
<th>used</th>
<th>load</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>pool_sales</td>
<td>2</td>
<td>1.2</td>
<td>8.3</td>
</tr>
<tr>
<td>0</td>
<td>pool_default</td>
<td>2</td>
<td>0.4</td>
<td>5.2</td>
</tr>
</tbody>
</table>

```
pset
```

<table>
<thead>
<tr>
<th>id</th>
<th>pool</th>
<th>size</th>
<th>used</th>
<th>load</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>pool_sales</td>
<td>2</td>
<td>1.4</td>
<td>8.4</td>
</tr>
<tr>
<td>0</td>
<td>pool_default</td>
<td>2</td>
<td>1.9</td>
<td>2.0</td>
</tr>
</tbody>
</table>

```
pset
```

<table>
<thead>
<tr>
<th>id</th>
<th>pool</th>
<th>size</th>
<th>used</th>
<th>load</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>pool_sales</td>
<td>2</td>
<td>1.1</td>
<td>8.0</td>
</tr>
<tr>
<td>0</td>
<td>pool_default</td>
<td>2</td>
<td>0.3</td>
<td>5.0</td>
</tr>
</tbody>
</table>

Reporting Resource Set Statistics

The following example uses the `poolstat` command with the `-r` option to report statistics for the processor set resource set. Note that the resource set `pset_default` is attached to more than one pool, so this processor set is listed once for each pool membership.

```
machine% poolstat -r pset
```

<table>
<thead>
<tr>
<th>id</th>
<th>pool</th>
<th>type</th>
<th>rid</th>
<th>rset</th>
<th>min</th>
<th>max</th>
<th>size</th>
<th>used</th>
<th>load</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>pool_default</td>
<td>pset</td>
<td>-1</td>
<td>pset_default</td>
<td>1</td>
<td>65K</td>
<td>2</td>
<td>1.2</td>
<td>8.3</td>
</tr>
<tr>
<td>6</td>
<td>pool_sales</td>
<td>pset</td>
<td>1</td>
<td>pset_sales</td>
<td>1</td>
<td>65K</td>
<td>2</td>
<td>1.2</td>
<td>8.3</td>
</tr>
<tr>
<td>2</td>
<td>pool_other</td>
<td>pset</td>
<td>-1</td>
<td>pset_default</td>
<td>1</td>
<td>10K</td>
<td>2</td>
<td>0.4</td>
<td>5.2</td>
</tr>
</tbody>
</table>
This chapter reviews the resource management framework and describes a hypothetical server consolidation project.

The following topics are covered in this chapter:

- “Configuration to Be Consolidated” on page 189
- “Consolidation Configuration” on page 190
- “Creating the Configuration” on page 190
- “Viewing the Configuration” on page 192

Configuration to Be Consolidated

In this example, five applications are being consolidated onto a single system. The target applications have resource requirements that vary, different user populations, and different architectures. Currently, each application exists on a dedicated server that is designed to meet the requirements of the application. The applications and their characteristics are identified in the following table.

<table>
<thead>
<tr>
<th>Application Description</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application server</td>
<td>Exhibits negative scalability beyond 2 CPUs</td>
</tr>
<tr>
<td>Database instance for application server</td>
<td>Heavy transaction processing</td>
</tr>
<tr>
<td>Application server in test and development environment</td>
<td>GUI-based, with untested code execution</td>
</tr>
<tr>
<td>Transaction processing server</td>
<td>Primary concern is response time</td>
</tr>
<tr>
<td>Standalone database instance</td>
<td>Processes a large number of transactions and serves multiple time zones</td>
</tr>
</tbody>
</table>
Consolidation Configuration

The following configuration is used to consolidate the applications onto a single system that has the resource pools and the dynamic resource pools facilities enabled.

- The application server has a two-CPU processor set.
- The database instance for the application server and the stand-alone database instance are consolidated onto a single processor set of at least four CPUs. The stand-alone database instance is guaranteed 75 percent of that resource.
- The test and development application server requires the IA scheduling class to ensure UI responsiveness. Memory limitations are imposed to lessen the effects of bad code builds.
- The transaction processing server is assigned a dedicated processor set of at least two CPUs, to minimize response latency.

This configuration covers known applications that are executing and consuming processor cycles in each resource set. Thus, constraints can be established that allow the processor resource to be transferred to sets where the resource is required.

- The `wt-load` objective is set to allow resource sets that are highly utilized to receive greater resource allocations than sets that have low utilization.
- The `locality` objective is set to `tight`, which is used to maximize processor locality.

An additional constraint to prevent utilization from exceeding 80 percent of any resource set is also applied. This constraint ensures that applications get access to the resources they require. Moreover, for the transaction processor set, the objective of maintaining utilization below 80 percent is twice as important as any other objectives that are specified. This importance will be defined in the configuration.

Creating the Configuration

Edit the `/etc/project` database file. Add entries to implement the required resource controls and to map users to resource pools, then view the file.

```
# cat /etc/project

user.app_server:2001:Production Application Server::project.pool=appserver_pool
user.app_db:2002:App Server DB::project.pool=db_pool;project.cpu-shares=(privileged,1,deny)
development:2003:Test and development::staff:project.pool=dev_pool;
process.max-address-space=(privileged,536870912,deny) keep with previous line
user.tp_engine:2004:Transaction Engine::project.pool=tp_pool
user.geo_db:2005:EDI DB::project.pool=db_pool;project.cpu-shares=(privileged,3,deny)
```

Consolidation Configuration

System Administration Guide: Virtualization Using the Solaris Operating System • October 2009 (Beta)
Note – The development team has to execute tasks in the development project because access for this project is based on a user’s group ID (GID).

Create an input file named pool.host, which will be used to configure the required resource pools. View the file.

```shell
# cat pool.host
create system host
create pset dev_pset (uint pset.min = 0; uint pset.max = 2)
create pset tp_pset (uint pset.min = 2; uint pset.max=8)
create pset db_pset (uint pset.min = 4; uint pset.max = 6)
create pset app_pset (uint pset.min = 1; uint pset.max = 2)
create pool dev_pool (string pool.scheduler="IA")
create pool appserver_pool (string pool.scheduler="TS")
create pool db_pool (string pool.scheduler="FSS")
create pool tp_pool (string pool.scheduler="TS")
associate pool dev_pool (pset dev_pset)
associate pool appserver_pool (pset app_pset)
associate pool db_pool (pset db_pset)
associate pool tp_pool (pset tp_pset)
modify system tester (string system.poold.objectives="wt-load")
modify pset dev_pset (string pset.poold.objectives="locality tight; utilization < 80")
modify pset tp_pset (string pset.poold.objectives="locality tight; 2: utilization < 80")
modify pset app_pset (string pset.poold.objectives="locality tight; utilization < 80")
```

Update the configuration using the pool.host input file.

```shell
# poolcfg -f pool.host
```

Make the configuration active.

```shell
# pooladm -c
```

The framework is now functional on the system.

Enable DRP.

```shell
# svcadm enable pools/dynamic:default
```
Viewing the Configuration

To view the framework configuration, which also contains default elements created by the system, type:

```bash
# pooladm
system host
  string system.comment
  int system.version 1
  boolean system.bind-default true
  int system.poold.pid 177916
  string system.poold.objectives wt-load

pool dev_pool
  int pool.sys_id 125
  boolean pool.default false
  boolean pool.active true
  int pool.importance 1
  string pool.comment
  string pool.scheduler IA
  pset  dev_pset

pool appserver_pool
  int pool.sys_id 124
  boolean pool.default false
  boolean pool.active true
  int pool.importance 1
  string pool.comment
  string pool.scheduler TS
  pset  app_pset

pool db_pool
  int pool.sys_id 123
  boolean pool.default false
  boolean pool.active true
  int pool.importance 1
  string pool.comment
  string pool.scheduler FSS
  pset  db_pset

pool tp_pool
  int pool.sys_id 122
  boolean pool.default false
  boolean pool.active true
  int pool.importance 1
  string pool.comment
  string pool.scheduler TS
  pset  tp_pset
```
pool pool_default
  int pool.sys_id 0
  boolean pool.default true
  boolean pool.active true
  int pool.importance 1
  string pool.comment
  string pool.scheduler TS

pset dev_pset
  int pset.sys_id 4
  string pset.units population
  boolean pset.default false
  uint pset.min 0
  uint pset.max 2
  string pset.comment
  boolean pset.escapable false
  uint pset.load 0
  uint pset.size 0
  string pset.poold.objectives locality tight; utilization < 80

pset tp_pset
  int pset.sys_id 3
  string pset.units population
  boolean pset.default false
  uint pset.min 2
  uint pset.max 0
  string pset.comment
  boolean pset.escapable false
  uint pset.load 0
  uint pset.size 0
  string pset.poold.objectives locality tight; 2: utilization < 80

cpu
  int cpu.sys_id 1
  string cpu.comment
  string cpu.status on-line

cpu
  int cpu.sys_id 2
  string cpu.comment
  string cpu.status on-line

pset db_pset
  int pset.sys_id 2
  string pset.units population
  boolean pset.default false
uint pset.min 4
uint pset.max 6
string pset.comment
boolean pset.escapable false
uint pset.load 0
uint pset.size 0
string pset.poold.objectives locality tight; utilization < 80

cpu
  int cpu.sys_id 3
  string cpu.comment
  string cpu.status on-line

cpu
  int cpu.sys_id 4
  string cpu.comment
  string cpu.status on-line

cpu
  int cpu.sys_id 5
  string cpu.comment
  string cpu.status on-line

cpu
  int cpu.sys_id 6
  string cpu.comment
  string cpu.status on-line

pset app_pset
  int pset.sys_id 1
  string pset.units population
  boolean pset.default false
  uint pset.min 1
  uint pset.max 2
  string pset.comment
  boolean pset.escapable false
  uint pset.load 0
  uint pset.size 0
  string pset.poold.objectives locality tight; utilization < 80

  cpu
    int cpu.sys_id 7
    string cpu.comment
    string cpu.status on-line

pset pset_default
  int pset.sys_id -1
  string pset.units population
  boolean pset.default true
  uint pset.min 1
A graphic representation of the framework follows.

Note – In the pool db_pool, the standalone database instance is guaranteed 75 percent of the CPU resource.
This part covers Solaris™ Zones software partitioning technology, which provides a means of virtualizing operating system services to create an isolated environment for running applications. This isolation prevents processes that are running in one zone from monitoring or affecting processes running in other zones.
The Solaris™ Zones facility in the Solaris Operating System provides an isolated environment in which to run applications on your system. Solaris Zones are a component of the Solaris Container environment.

This chapter provides limited information about the ipkg brand for the OpenSolaris 2006.09 release. This design is under development, and the solution for OpenSolaris might change.

The chapter also covers the following general zones topics:
- “Zones Overview” on page 201
- “When to Use Zones” on page 202
- “How Zones Work” on page 204
- “Features Provided by Non-Global Zones” on page 210
- “Setting Up Zones on Your System (Task Map)” on page 211

If you are ready to start creating zones on your system, skip to Chapter 16, “Non-Global Zone Configuration (Overview).”

Note – For information on branded zones, see Part III.

For information on using zones on a Solaris Trusted Extensions system, see Chapter 16, “Managing Zones in Trusted Extensions (Tasks),” in Solaris Trusted Extensions Administrator’s Procedures and “Zones in Trusted Extensions” in Solaris Trusted Extensions Transition Guide.
About Zones in the OpenSolaris 2009.06 Release

Zones on the OpenSolaris release are a work in progress; they are evolving as development continues. The software management aspect of zones is changing because OpenSolaris uses IPS instead of SVr4 packaging and patching. Thus, work is still being done where zones are impacted by software management, such as installation and migration.

The Image Packaging System (IPS) is a new model for software management in the OpenSolaris 2009.06 Release, and zones are changing to utilize this model. The following differences should be noted:

- The `ipkg` brand is the default instead of the native brand, which is the default on SX systems.
- `ipkg` branded zones are whole-root type only; `inherit-pkg-dir` should not be used.
  The sparse root type of zone describes a fundamental interaction between zones and the package management system, and IPS doesn’t support this concept. Sun is working on providing the positive attributes of sparse root zones in other ways.
- Zones have different software management related functionality in these areas:
  - IPS versus SVR4 packaging
  - Install, detach/attach, “physical to virtual” (P2V) capability
- Zones have different global zone software operations; they employ manual syncing, not patching. Currently, the zones don’t automatically update when you `pkg image-update` the system. You must manually update the zones after rebooting to keep them in sync with the global zone. In a future release, this update should happen automatically.

Note – Until `pkg_image-update` is fully supported, you can use `zoneadm detach` and `attach -u` as a workaround. Detach the zone before running `pkg_image-update`, then use `attach -u` after running `pkg_image-update`. See “About Migrating a Zone” on page 306 for more information on these commands.

- Global zones are integrated with `beadm` and use boot environments as described in `beadm Zones Support` (http://dlc.sun.com/osol/docs/content/2008.11/snapupgrade/ zones.html). In a future release, support for `beadm` inside zones is planned.
- The zone root is a ZFS™ dataset. In a future release, this feature will enable support for `beadm` inside zones for `pkg_image-update`, just as you can do in the global zone. To accomplish this, the zone’s root dataset must be controlled inside the zone.
- Zone software is minimized to start; any additional packages the zone requires must be added. See OpenSolaris Package Repositories (http://www.opensolaris.org/os/package) for more information.
You must be on the network to install a zone. Installation is from the OpenSolaris Packaging Repository. See OpenSolaris 2009.06 Image Packaging System Guide (http://dlc.sun.com/osol/docs/content/2009.06/IMGPACKAGESYS/) for more information.

To control the software installed in the ipkg brand zone, use the -e option to the zoneadm install command.

```
# zoneadm -z zonename install -e pkgA -e pkgB ...
```

For information on zones in the OpenSolaris release, visit the Zones OpenSolaris Community (http://www.opensolaris.org/os/community/zones). This site will be updated to address the latest issues.

Zones Overview

The Solaris Zones partitioning technology is used to virtualize operating system services and provide an isolated and secure environment for running applications. A zone is a virtualized operating system environment created within a single instance of the Solaris Operating System. When you create a zone, you produce an application execution environment in which processes are isolated from the rest of the system. This isolation prevents processes that are running in one zone from monitoring or affecting processes that are running in other zones. Even a process running with superuser credentials cannot view or affect activity in other zones.

A zone also provides an abstract layer that separates applications from the physical attributes of the machine on which they are deployed. Examples of these attributes include physical device paths.

Zones can be used on any machine that is running the Solaris 10 or later Solaris release. The upper limit for the number of zones on a system is 8192. The number of zones that can be effectively hosted on a single system is determined by the total resource requirements of the application software running in all of the zones.

There are two types of non-global zone root file system models: sparse and whole root.

- Sparse root zones are native branded zones. The sparse root zone model optimizes the sharing of objects. Note that some brands, especially the ipkg brand, do not support the sparse root model.

- The whole root zone model provides the maximum configurability.

These concepts are discussed in Chapter 17, “Planning and Configuring Non-Global Zones (Tasks)”.

Zones Overview
About Branded Zones

Branded zones (BrandZ) provide the framework to create containers that contain alternative sets of runtime behaviors. *Brand* can refer to a wide range of operating environments. For example, the non-global zone can emulate the Solaris 8 Operating System, or an operating environment such as Linux. *Every zone* is configured with an associated brand.

The brand defines the operating environment that can be installed in the zone and determines how the system will behave within the zone so that the software installed in the zone functions correctly. In addition, a zone’s brand is used to identify the correct application type at application launch time. All branded zone management is performed through extensions to the standard zones structure. Most administration procedures are identical for all zones.

BrandZ extends the zones tools in the following ways:

- The `zonecfg` command is used to set a zone’s brand type when the zone is configured.
- The `zoneadm` command is used to report a zone’s brand type as well as administer the zone.

Although you can configure and install branded zones on a Solaris Trusted Extensions system that has labels enabled, you cannot boot branded zones on this system configuration, *unless* the brand being booted is the labeled brand on an OpenSolaris system configuration.

When to Use Zones

Zones are ideal for environments that consolidate a number of applications on a single server. The cost and complexity of managing numerous machines make it advantageous to consolidate several applications on larger, more scalable servers.

The following figure shows a system with four zones. Each of the zones apps, users, and work is running a workload unrelated to the workloads of the other zones, in a sample consolidated environment. This example illustrates that different versions of the same application can be run without negative consequences in different zones, to match the consolidation requirements. Each zone can provide a customized set of services.
Zones enable more efficient resource utilization on your system. Dynamic resource reallocation permits unused resources to be shifted to other containers as needed. Fault and security isolation mean that poorly behaved applications do not require a dedicated and under-utilized system. With the use of zones, these applications can be consolidated with other applications.

Zones allow you to delegate some administrative functions while maintaining overall system security.
How Zones Work

A non-global zone can be thought of as a box. One or more applications can run in this box without interacting with the rest of the system. Solaris zones isolate software applications or services by using flexible, software-defined boundaries. Applications that are running in the same instance of the Solaris Operating System can then be managed independently of one another. Thus, different versions of the same application can be run in different zones, to match the requirements of your configuration.

A process assigned to a zone can manipulate, monitor, and directly communicate with other processes that are assigned to the same zone. The process cannot perform these functions with processes that are assigned to other zones in the system or with processes that are not assigned to a zone. Processes that are assigned to different zones are only able to communicate through network APIs.

IP networking can be configured in two different ways, depending on whether the zone has its own exclusive IP instance or shares the IP layer configuration and state with the global zone. For more information about IP types in zones, see “Zone Network Interfaces” on page 219. For configuration information, see "How to Configure the Zone” on page 249.

Every Solaris system contains a global zone. The global zone has a dual function. The global zone is both the default zone for the system and the zone used for system-wide administrative control. All processes run in the global zone if no non-global zones, referred to simply as zones, are created by the global administrator.

The global zone is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled. Only the global zone is bootable from the system hardware. Administration of the system infrastructure, such as physical devices, routing in a shared-IP zone, or dynamic reconfiguration (DR), is only possible in the global zone. Appropriately privileged processes running in the global zone can access objects associated with other zones.

Unprivileged processes in the global zone might be able to perform operations not allowed to privileged processes in a non-global zone. For example, users in the global zone can view information about every process in the system. If this capability presents a problem for your site, you can restrict access to the global zone.

Each zone, including the global zone, is assigned a zone name. The global zone always has the name global. Each zone is also given a unique numeric identifier, which is assigned by the system when the zone is booted. The global zone is always mapped to ID 0. Zone names and numeric IDs are discussed in “Using the zonecfg Command” on page 225.

Each zone also has a node name that is completely independent of the zone name. The node name is assigned by the administrator of the zone. For more information, see “Non-Global Zone Node Name” on page 353.

Each zone has a path to its root directory that is relative to the global zone’s root directory. For more information, see “Using the zonecfg Command” on page 225.
The scheduling class for a non-global zone is set to the scheduling class for the system by default. See “Scheduling Class” on page 218 for a discussion of methods used to set the scheduling class in a zone.

## Summary of Zone Features

The following table summarizes the characteristics of global and non-global zones.

<table>
<thead>
<tr>
<th>Type of Zone</th>
<th>Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>▪ Is assigned ID 0 by the system</td>
</tr>
<tr>
<td></td>
<td>▪ Provides the single instance of the Solaris kernel that is bootable and running on the system</td>
</tr>
<tr>
<td></td>
<td>▪ Contains a complete installation of the Solaris system software packages</td>
</tr>
<tr>
<td></td>
<td>▪ Can contain additional software packages or additional software, directories, files, and other data not installed through packages</td>
</tr>
<tr>
<td></td>
<td>▪ Provides a complete and consistent product database that contains information about all software components installed in the global zone</td>
</tr>
<tr>
<td></td>
<td>▪ Holds configuration information specific to the global zone only, such as the global zone host name and file system table</td>
</tr>
<tr>
<td></td>
<td>▪ Is the only zone that is aware of all devices and all file systems</td>
</tr>
<tr>
<td></td>
<td>▪ Is the only zone with knowledge of non-global zone existence and configuration</td>
</tr>
<tr>
<td></td>
<td>▪ Is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled</td>
</tr>
</tbody>
</table>
### How Non-Global Zones Are Administered

A global administrator has superuser privileges or the Primary Administrator role. When logged into the global zone, the global administrator can monitor and control the system as a whole.

A non-global zone can be administered by a zone administrator. The global administrator assigns the Zone Management profile to the zone administrator. The privileges of a zone administrator are confined to a non-global zone.

### How Non-Global Zones Are Created

The global administrator uses the `zonecfg` command to configure a zone by specifying various parameters for the zone’s virtual platform and application environment. The zone is then installed by the global administrator, who uses the zone administration command `zoneadm` to install software at the package level into the file system hierarchy established for the zone. The `zoneadm` command is used to boot the zone. The global administrator can then log in to the installed zone by using the `zlogin` command. At first login, the internal configuration for the zone is completed.
Non-Global Zone State Model

A non-global zone can be in one of the following six states:

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configured</td>
<td>The zone's configuration is complete and committed to stable storage. However, those elements of the zone's application environment that must be specified after initial boot are not yet present.</td>
</tr>
<tr>
<td>Incomplete</td>
<td>During an install or uninstall operation, zoneadm sets the state of the target zone to incomplete. Upon successful completion of the operation, the state is set to the correct state.</td>
</tr>
<tr>
<td>Installed</td>
<td>The zone's configuration is instantiated on the system. The zoneadm command is used to verify that the configuration can be successfully used on the designated Solaris system. Packages are installed under the zone's root path. In this state, the zone has no associated virtual platform.</td>
</tr>
<tr>
<td>Ready</td>
<td>The virtual platform for the zone is established. The kernel creates the zsched process, network interfaces are set up and made available to the zone, file systems are mounted, and devices are configured. A unique zone ID is assigned by the system. At this stage, no processes associated with the zone have been started.</td>
</tr>
<tr>
<td>Running</td>
<td>User processes associated with the zone application environment are running. The zone enters the running state as soon as the first user process associated with the application environment (init) is created.</td>
</tr>
<tr>
<td>Shutting down and Down</td>
<td>These states are transitional states that are visible while the zone is being halted. However, a zone that is unable to shut down for any reason will stop in one of these states.</td>
</tr>
</tbody>
</table>
Chapter 19, "Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks)," and the `zoneadm(1M)` man page describe how to use the `zoneadm` command to initiate transitions between these states.

**TABLE 15–1  Commands That Affect Zone State**

<table>
<thead>
<tr>
<th>Current Zone State</th>
<th>Applicable Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configured</td>
<td><code>zonecfg -z zonename verify</code>&lt;br&gt;<code>zonecfg -z zonename commit</code>&lt;br&gt;<code>zonecfg -z zonename delete</code>&lt;br&gt;<code>zoneadm -z zonename attach</code>&lt;br&gt;<code>zoneadm -z zonename verify</code>&lt;br&gt;<code>zoneadm -z zonename install</code>&lt;br&gt;<code>zoneadm -z zonename clone</code>&lt;br&gt;You can also use <code>zonecfg</code> to rename a zone in the configured or installed state.</td>
</tr>
<tr>
<td>Incomplete</td>
<td><code>zoneadm -z zonename uninstall</code></td>
</tr>
<tr>
<td>Installed</td>
<td><code>zoneadm -z zonename ready</code> (optional)&lt;br&gt;<code>zoneadm -z zonename boot</code>&lt;br&gt;<code>zoneadm -z zonename uninstall</code> uninstalls the configuration of the specified zone from the system.&lt;br&gt;<code>zoneadm -z zonename move path</code>&lt;br&gt;<code>zoneadm -z zonename detach</code>&lt;br&gt;<code>zonecfg -z zonename</code> can be used to add or remove an <code>attr</code>, <code>bootargs</code>, <code>capped-memory</code>, <code>dataset</code>, <code>capped-cpu</code>, <code>dedicated-cpu</code>, <code>device</code>, <code>fs</code>, <code>ip-type</code>, <code>limitpriv</code>, <code>net</code>, <code>rctl</code>, or <code>scheduling-class</code> property. You can also rename a zone in the installed state. The <code>inherit-pkg-dir</code> resources cannot be changed.</td>
</tr>
<tr>
<td>Ready</td>
<td><code>zoneadm -z zonename boot</code>&lt;br&gt;<code>zoneadm halt</code> and system reboot return a zone in the ready state to the installed state.&lt;br&gt;<code>zonecfg -z zonename</code> can be used to add or remove <code>attr</code>, <code>bootargs</code>, <code>capped-memory</code>, <code>dataset</code>, <code>capped-cpu</code>, <code>dedicated-cpu</code>, <code>device</code>, <code>fs</code>, <code>ip-type</code>, <code>limitpriv</code>, <code>net</code>, <code>rctl</code>, or <code>scheduling-class</code> property. The <code>inherit-pkg-dir</code> resources cannot be changed.</td>
</tr>
</tbody>
</table>
### TABLE 15–1  Commands That Affect Zone State  (Continued)

<table>
<thead>
<tr>
<th>Current Zone State</th>
<th>Applicable Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td><code>zlogin</code> <em>options</em> <code>zonename</code></td>
</tr>
<tr>
<td></td>
<td><code>zoneadm -z zonename</code> <code>reboot</code></td>
</tr>
<tr>
<td></td>
<td><code>zoneadm -z zonename</code> <code>halt</code> returns a ready zone to the installed state.</td>
</tr>
<tr>
<td></td>
<td><code>zoneadm</code> <code>halt</code> and system reboot return a zone in the running state to the installed state.</td>
</tr>
<tr>
<td></td>
<td><code>zonecfg -z zonename</code> can be used to add or remove an <code>attr</code>, <code>bootargs</code>, <code>capped-memory</code>, <code>dataset</code>, <code>capped-cpu</code>, <code>dedicated-cpu</code>, <code>device</code>, <code>fs</code>, <code>ip-type</code>, <code>limitpriv</code>, <code>net</code>, <code>rctl</code>, or <code>scheduling-class</code> property. The <code>zonepath</code> and <code>inherit-pkg-dir</code> resources cannot be changed.</td>
</tr>
</tbody>
</table>

---

**Note** – Parameters changed through `zonecfg` do not affect a running zone. The zone must be rebooted for the changes to take effect.

### Non-Global Zone Characteristics

A zone provides isolation at almost any level of granularity you require. A zone does not need a dedicated CPU, a physical device, or a portion of physical memory. These resources can either be multiplexed across a number of zones running within a single domain or system, or allocated on a per-zone basis using the resource management features available in the operating system.

Each zone can provide a customized set of services. To enforce basic process isolation, a process can see or signal only those processes that exist in the same zone. Basic communication between zones is accomplished by giving each zone IP network connectivity. An application running in one zone cannot observe the network traffic of another zone. This isolation is maintained even though the respective streams of packets travel through the same physical interface.

Each zone is given a portion of the file system hierarchy. Because each zone is confined to its subtree of the file system hierarchy, a workload running in a particular zone cannot access the on-disk data of another workload running in a different zone.

Files used by naming services reside within a zone’s own root file system view. Thus, naming services in different zones are isolated from one another and the services can be configured differently.
Using Resource Management Features With Non-Global Zones

If you use resource management features, you should align the boundaries of the resource management controls with those of the zones. This alignment creates a more complete model of a virtual machine, where namespace access, security isolation, and resource usage are all controlled.

Any special requirements for using the various resource management features with zones are addressed in the individual chapters of this manual that document those features.

Features Provided by Non-Global Zones

Non-global zones provide the following features:

Security Once a process has been placed in a zone other than the global zone, neither the process nor any of its subsequent children can change zones.

Network services can be run in a zone. By running network services in a zone, you limit the damage possible in the event of a security violation. An intruder who successfully exploits a security flaw in software running within a zone is confined to the restricted set of actions possible within that zone. The privileges available within a zone are a subset of those available in the system as a whole.

Isolation Zones allow the deployment of multiple applications on the same machine, even if those applications operate in different trust domains, require exclusive access to a global resource, or present difficulties with global configurations. For example, multiple applications running in different shared-IP zones on the same system can bind to the same network port by using the distinct IP addresses associated with each zone or by using the wildcard address. The applications are also prevented from monitoring or intercepting each other’s network traffic, file system data, or process activity.

Network Isolation If a zone needs to be isolated at the IP layer on the network, for example, by being connected to different VLANs or different LANs than the global zone and other non-global zones, then for security reasons the zone can have an exclusive IP. The exclusive-IP zone can be used to consolidate applications that must communicate on different subnets that are on different VLANs or different LANs.

Zones can also be configured as shared-IP zones. These zones connect to the same VLANs or same LANs as the global zone and share the IP
routing configuration with the global zone. Shared-IP zones have separate IP addresses, but share the other parts of IP.

Virtualization Zones provide a virtualized environment that can hide details such as physical devices and the system’s primary IP address and host name from applications. The same application environment can be maintained on different physical machines. The virtualized environment allows separate administration of each zone. Actions taken by a zone administrator in a non-global zone do not affect the rest of the system.

Granularity A zone can provide isolation at almost any level of granularity. See “Non-Global Zone Characteristics” on page 209 for more information.

Environment Zones do not change the environment in which applications execute except when necessary to achieve the goals of security and isolation. Zones do not present a new API or ABI to which applications must be ported. Instead, zones provide the standard Solaris interfaces and application environment, with some restrictions. The restrictions primarily affect applications that attempt to perform privileged operations.

Applications in the global zone run without modification, whether or not additional zones are configured.

**Setting Up Zones on Your System (Task Map)**

The following table provides a basic overview of the tasks that are involved in setting up zones on your system for the first time.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify the applications that you would like to run in zones.</td>
<td>Review the applications running on your system:  ■ Determine which applications are critical to your business goals.  ■ Assess the system needs of the applications you are running.</td>
<td>Refer to your business goals and to your system documentation if necessary.</td>
</tr>
<tr>
<td>Task</td>
<td>Description</td>
<td>For Instructions</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Determine how many zones to configure.</td>
<td>Assess:</td>
<td>See “Evaluating the Current System Setup” on page 244.</td>
</tr>
<tr>
<td></td>
<td>■ The performance requirements of the applications you intend to run in zones</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ The availability of the recommended 100 MB of free disk space per zone to be installed</td>
<td></td>
</tr>
<tr>
<td>Determine whether you will use resource pools with your zone to create a container.</td>
<td>If you are also using resource management features on your system, align the zones with the resource management boundaries. Configure resource pools before you configure zones. Note that you can add zone-wide resource controls and pool functionality to a zone quickly by using zonecfg properties.</td>
<td>See “How to Configure the Zone” on page 249, and Chapter 13, “Creating and Administering Resource Pools (Tasks).”</td>
</tr>
<tr>
<td>Perform the preconfiguration tasks.</td>
<td>Determine the zone name and the zone path. Determine whether the zone will be a shared-IP zone or an exclusive-IP zone, and obtain IP addresses or the data-link name. Determine the required file systems and devices for each zone. Determine the scheduling class for the zone. Determine the set of privileges that processes inside the zone should be limited to, if the standard default set is not sufficient. Note that some zonecfg settings automatically add privileges. For example, <code>ip -type=exclusive</code> automatically adds multiple privileges required to configure and manage network stacks.</td>
<td>For information on the zone name and path, IP types, IP addresses, file systems, devices, scheduling class, and privileges, see Chapter 16, “Non-Global Zone Configuration (Overview),” and “Evaluating the Current System Setup” on page 244. For a listing of default privileges and privileges that can be configured in a non-global zone, see “Privileges in a Non-Global Zone” on page 367. For information about IP feature availability, see “Networking in Shared-IP Non-Global Zones” on page 360 and “Networking in Exclusive-IP Non-Global Zones” on page 362.</td>
</tr>
<tr>
<td>Develop configurations.</td>
<td>Configure non-global zones.</td>
<td>See “Configuring, Verifying, and Committing a Zone” on page 249 and the zonecfg(1M) man page.</td>
</tr>
</tbody>
</table>
### Setting Up Zones on Your System (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>As global administrator, verify and install configured zones.</td>
<td>Zones must be verified and installed prior to login.</td>
<td>See Chapter 18, &quot;About Installing, Halting, Uninstalling, and Cloning Non-Global Zones (Overview),&quot; and Chapter 19, &quot;Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks).&quot;</td>
</tr>
<tr>
<td>As global administrator, boot the non-global zones.</td>
<td>Boot each zone to place the zone in the running state.</td>
<td>See Chapter 18, &quot;About Installing, Halting, Uninstalling, and Cloning Non-Global Zones (Overview),&quot; and Chapter 19, &quot;Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks).&quot;</td>
</tr>
<tr>
<td>As global administrator, perform the initial internal configuration of the zone.</td>
<td>Place a <code>sysidcfg</code> file in the zone’s <code>/etc</code> directory or log in to each non-global zone using the <code>zlogin</code> command with the <code>-C</code> option and enter the requested information, including assigning the zone root password.</td>
<td>See Chapter 20, &quot;Non-Global Zone Login (Overview),&quot; and Chapter 21, &quot;Logging In to Non-Global Zones (Tasks).&quot;</td>
</tr>
<tr>
<td>Prepare the new zone for production use.</td>
<td>Create user accounts, add additional software, and customize the zone’s configuration.</td>
<td>Refer to the documentation you use to set up a newly installed machine. Special considerations applicable to a system with zones installed are covered in this guide.</td>
</tr>
</tbody>
</table>
Non-Global Zone Configuration (Overview)

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

The following topics are covered in this chapter:

- “About Resources in Zones” on page 215
- “Pre-Installation Configuration Process” on page 216
- “Zone Components” on page 216
- “Using the zonecfg Command” on page 225
- “zonecfg Modes” on page 226
- “Zone Configuration Data” on page 229
- “Tecla Command-Line Editing Library” on page 238

After you have learned about zone configuration, go to Chapter 17, “Planning and Configuring Non-Global Zones (Tasks),” to configure non-global zones for installation on your system.

For information about \( \times \) branded zone configuration, see Chapter 30, “Planning the \( \times \) Branded Zone Configuration (Overview),” and Chapter 31, “Configuring the \( \times \) Branded Zone (Tasks).”

About Resources in Zones

A zone that includes resource management features is called a container. Resources that can be controlled in a container include the following:

- Resource pools or assigned CPUs, which are used for partitioning machine resources.
- Resource controls, which provide a mechanism for the constraint of system resources.
- Scheduling class, which enables you to control the allocation of available CPU resources among zones, based on their importance. This importance is expressed by the number of shares of CPU resources that you assign to each zone.
Pre-Installation Configuration Process

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

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

- Ensures that a zone path is specified
- Ensures that all of the required properties for each resource are specified

For more information about the `zonecfg` command, see the `zonecfg(1M)` man page.

Zone Components

This section covers the required and optional zone components that can be configured. Additional information is provided in "Zone Configuration Data" on page 229.

Zone Name and Path

You must choose a name and a path for your zone.

Zone Autoboot

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

Resource Pool Association

If you have configured resource pools on your system as described in Chapter 13, "Creating and Administering Resource Pools (Tasks)," you can use the `pool` property to associate the zone with one of the resource pools when you configure the zone.

If you do not have resource pools configured, you can still specify that a subset of the system's processors be dedicated to a non-global zone while it is running by using the `dedicated-cpu` resource. The system will dynamically create a temporary pool for use while the zone is running. With specification through `zonecfg`, pool settings propagate during migrations.
Note – A zone configuration using a persistent pool set through the pool\ property is incompatible with a temporary pool configured through the dedicated-cpu resource. You can set only one of these two properties.

**dedicated-cpu Resource**

The dedicated-cpu resource specifies that a subset of the system's processors should be dedicated to a non-global zone while it is running. When the zone boots, the system will dynamically create a temporary pool for use while the zone is running.

With specification in zonecfg, pool settings propagate during migrations.

The dedicated-cpu resource sets limits for ncpus, and optionally, importance.

**ncpus** Specify the number of CPUs or specify a range, such as 2–4 CPUs. If you specify a range because you want dynamic resource pool behavior, also do the following:

- Set the importance property.
- Enable the poold service. For instructions, see “How to Enable the Dynamic Resource Pools Service Using svcadm” on page 168.

**importance** If you are using a CPU range to achieve dynamic behavior, also set the importance property. The importance property, which is optional, defines the relative importance of the pool. This property is only needed when you specify a range for ncpus and are using dynamic resource pools managed by poold. If poold is not running, then importance is ignored. If poold is running and importance is not set, importance defaults to 1. For more information, see “pool.importance Property Constraint” on page 153.

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

**capped-cpu Resource**

The capped-cpu resource provides an absolute fine-grained limit on the amount of CPU resources that can be consumed by a project or a zone. When used in conjunction with processor sets, CPU caps limit CPU usage within a set. The capped-cpu resource has a single ncpus property that is a positive decimal with two digits to the right of the decimal. This property corresponds to units of CPUs. The resource does not accept a range. The resource
does accept a decimal number. When specifying ncpus, a value of 1 means 100 percent of a CPU. A value of 1.25 means 125 percent, because 100 percent corresponds to one full CPU on the system.

**Note** – The capped-cpu resource and the dedicated-cpu resource are incompatible.

### Scheduling Class

You can use the *fair share scheduler* (FSS) to control the allocation of available CPU resources among zones, based on their importance. This importance is expressed by the number of shares of CPU resources that you assign to each zone. Even if you are not using FSS to manage CPU resource allocation between zones, you can set the zone’s scheduling-class to use FSS so that you can set shares on projects within the zone.

When you explicitly set the cpu-shares property, the fair share scheduler (FSS) will be used as the scheduling class for that zone. However, the preferred way to use FSS in this case is to set FSS to be the system default scheduling class with the dispadmind command. That way, all zones will benefit from getting a fair share of the system CPU resources. If cpu-shares is not set for a zone, the zone will use the system default scheduling class. The following actions set the scheduling class for a zone:

- You can use the scheduling-class property in zonecfg to set the scheduling class for the zone.
- You can set the scheduling class for a zone through the resource pools facility. If the zone is associated with a pool that has its pool.scheduler property set to a valid scheduling class, then processes running in the zone run in that scheduling class by default. See “Introduction to Resource Pools” on page 144 and “How to Associate a Pool With a Scheduling Class” on page 175.
- If the cpu-shares rctl is set and FSS has not been set as the scheduling class for the zone through another action, zoneadm sets the scheduling class to FSS when the zone boots.
- If the scheduling class is not set through any other action, the zone inherits the system default scheduling class.

Note that you can use the priocntl described in the priocntl(1) man page to move running processes into a different scheduling class without changing the default scheduling class and rebooting.
Physical Memory Control and the capped-memory Resource

The capped-memory resource sets limits for physical, swap, and locked memory. Each limit is optional, but at least one must be set.

- Determine values for this resource if you plan to cap memory for the zone by using rcapd from the global zone. The physical property of the capped-memory resource is used by rcapd as the max-rss value for the zone.
- The swap property of the capped-memory resource is the preferred way to set the zone.max-swap resource control.
- The locked property of the capped-memory resource is the preferred way to set the zone.max-locked-memory resource control.

Note – Applications generally do not lock significant amounts of memory, but you might decide to set locked memory if the zone's applications are known to lock memory. If zone trust is a concern, you can also consider setting the locked memory cap to 10 percent of the system's physical memory, or 10 percent of the zone's physical memory cap.

For more information, see Chapter 10, “Physical Memory Control Using the Resource Capping Daemon (Overview),” Chapter 11, “Administering the Resource Capping Daemon (Tasks),” and “How to Configure the Zone” on page 249. To temporarily set a resource cap for a zone, see “How to Specify a Temporary Resource Cap for a Zone” on page 137.

Zone Network Interfaces

Zone network interfaces configured by the zonecfg command to provide network connectivity will automatically be set up and placed in the zone when it is booted.

The Internet Protocol (IP) layer accepts and delivers packets for the network. This layer includes IP routing, the Address Resolution Protocol (ARP), IP security architecture (IPsec), and IP Filter.

There are two IP types available for non-global zones, shared-IP and exclusive-IP. The shared-IP zone shares a network interface and the exclusive-IP zone must have a dedicated network interface.

For information about IP features in each type, see “Networking in Shared-IP Non-Global Zones” on page 360 and “Networking in Exclusive-IP Non-Global Zones” on page 362.
**Shared-IP Non-Global Zones**

The shared-IP zone is the default type. The zone must have one or more dedicated IP addresses. A shared-IP zone shares the IP layer configuration and state with the global zone. The zone should use the shared-IP instance if both of the following are true:

- The zone is to be connected to the same data-link, that is, be on the same IP subnet or subnets as the global zone
- You do not want the other capabilities that the exclusive-IP zone provides.

Shared-IP zones are assigned one or more IP addresses using the `zonecfg` command. The data-link names must also be configured in the global zone.

In the `zonecfg net` resource, the `address` and the `physical` properties must be set. The `defrouter` property is optional.

These addresses are associated with logical network interfaces. The `ifconfig` command can be used from the global zone to add or remove logical interfaces in a running zone. For more information, see "Shared-IP Network Interfaces" on page 361.

**Exclusive-IP Non-Global Zones**

Full IP-level functionality is available in an exclusive-IP zone.

An exclusive-IP zone has its own IP-related state.

This includes the ability to use the following features in an exclusive-IP zone:

- DHCPv4 and IPv6 stateless address autoconfiguration
- IP Filter, including network address translation (NAT) functionality
- IP Network Multipathing (IPMP)
- IP routing
- `ndd` for setting TCP/UDP/SCTP as well as IP/ARP-level knobs
- IP security (IPsec) and Internet Key Exchange (IKE), which automates the provision of authenticated keying material for IPsec security association

An exclusive-IP zone is assigned its own set of data-links using the `zonecfg` command. The zone is given a data-link name such as `xge0`, `e1000g1`, or `bge32001`, using the `physical` property of the `net` resource. The `address` and the `defrouter` properties of the `net` resource are not set.

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

The `dladm` command can be used with the `show-linkprop` subcommand to show the assignment of data-links to running exclusive-IP zones. The `dladm` command can be used with
the set - linkprop subcommand to assign additional data-links to running zones. See “Administering Data-Links in Exclusive-IP Non-Global Zones” on page 395 for usage examples.

Inside a running exclusive-IP zone, the ifconfig command can be used to configure IP, which includes the ability to add or remove logical interfaces. The IP configuration in a zone can be set up in the same way as for the global zone, by using the sysidtools described in sysidcfg(4).

Note – The IP configuration of an exclusive-IP zone can only be viewed from the global zone by using the zlogin command. An example follows.

global# zlogin zone1 ifconfig -a

Security Differences Between Shared-IP and Exclusive-IP Non-Global Zones

In a shared-IP zone, applications in the zone, including the superuser, cannot send packets with source IP addresses other than the ones assigned to the zone through the zonecfg utility. This type of zone does not have access to send and receive arbitrary data-link (layer 2) packets.

For an exclusive-IP zone, zonecfg instead grants the entire specified data-link to the zone. As a result, the superuser in an exclusive-IP zone can send spoofed packets on those data-links, just as can be done in the global zone.

Using Shared-IP and Exclusive-IP Non-Global Zones at the Same Time

The shared-IP zones always share the IP layer with the global zone, and the exclusive-IP zones always have their own instance of the IP layer. Both shared-IP zones and exclusive-IP zones can be used on the same machine.

File Systems Mounted in Zones

Generally, the file systems mounted in a zone include the following:

- The set of file systems mounted when the virtual platform is initialized
- The set of file systems mounted from within the application environment itself

This can include, for example, the following file systems:

- File systems specified in a zone’s /etc/vfstab file
- AutoFS and AutoFS-triggered mounts
- Mounts explicitly performed by a zone administrator
Certain restrictions are placed on mounts performed from within the application environment. These restrictions prevent the zone administrator from denying service to the rest of the system, or otherwise negatively impacting other zones.

There are security restrictions associated with mounting certain file systems from within a zone. Other file systems exhibit special behavior when mounted in a zone. See “File Systems and Non-Global Zones” on page 353 for more information.

**Host ID in Zones**

You can set a hostid property for the non-global zone that is different from the hostid of the global zone. This would be done, for example, in the case of a machine migrated into a zone on another system. Applications now inside the zone might depend on the original hostid. See “Resource and Property Types” on page 229 for more information.

**Configured Devices in Zones**

The zonecfg command uses a rule-matching system to specify which devices should appear in a particular zone. Devices matching one of the rules are included in the zone’s /dev file system. For more information, see “How to Configure the Zone” on page 249.

**Setting Zone-Wide Resource Controls**

The global administrator can set privileged zone-wide resource controls for a zone. Zone-wide resource controls limit the total resource usage of all process entities within a zone.

These limits are specified for both the global and non-global zones by using the zonecfg command. See “How to Configure the Zone” on page 249.

The preferred, simpler method for setting a zone-wide resource control is to use the property name instead of the rctl resource.

The zone.cpu-cap resource control sets an absolute limit on the amount of CPU resources that can be consumed by a zone. A value of 100 means 100 percent of one CPU as the project.cpu-cap setting. A value of 125 is 125 percent, because 100 percent corresponds to one full CPU on the system when using CPU caps.

---

**Note** – When setting the capped-cpu resource, you can use a decimal number for the unit. The value correlates to the zone.capped-cpu resource control, but the setting is scaled down by 100. A setting of 1 is equivalent to a setting of 100 for the resource control.
The `zone.cpu-shares` resource control sets a limit on the number of fair share scheduler (FSS) CPU shares for a zone. CPU shares are first allocated to the zone, and then further subdivided among projects within the zone as specified in the `project.cpu-shares` entries. For more information, see "Using the Fair Share Scheduler on a Solaris System With Zones Installed" on page 397. The global property name for this control is `cpu-shares`.

The `zone.max-locked-memory` resource control limits the amount of locked physical memory available to a zone. The allocation of the locked memory resource across projects within the zone can be controlled by using the `project.max-locked-memory` resource control. See Table 6–1 for more information.

The `zone.max-lwps` resource control enhances resource isolation by preventing too many LWP s in one zone from affecting other zones. The allocation of the LWP resource across projects within the zone can be controlled by using the `project.max-lwps` resource control. See Table 6–1 for more information. The global property name for this control is `max-lwps`.

The `zone.max-msg-ids`, `zone.max-sem-ids`, `zone.max-shm-ids`, and `zone.max-shm-memory` resource controls are used to limit System V resources used by all processes within a zone. The allocation of System V resources across projects within the zone can be controlled by using the project versions of these resource controls. The global property names for these controls are `max-msg-ids`, `max-sem-ids`, `max-shm-ids`, and `max-shm-memory`.

The `zone.max-swap` resource control limits swap consumed by user process address space mappings and tmpfs mounts within a zone. The output of `prstat -Z` displays a SWAP column. The swap reported is the total swap consumed by the zone’s processes and tmpfs mounts. This value assists in monitoring the swap reserved by each zone, which can be used to choose an appropriate `zone.max-swap` setting.

<table>
<thead>
<tr>
<th>Control Name</th>
<th>Global Property Name</th>
<th>Description</th>
<th>Default Unit</th>
<th>Value Used For</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>zone.cpu-cap</code></td>
<td></td>
<td>Absolute limit on the amount of CPU resources for this zone</td>
<td>Quantity (number of CPUs), expressed as a percentage</td>
<td>Note – When setting as the capped-cpu resource, you can use a decimal number for the unit.</td>
</tr>
</tbody>
</table>

TABLE 16–1 Zone-Wide Resource Controls
<table>
<thead>
<tr>
<th>Control Name</th>
<th>Global Property Name</th>
<th>Description</th>
<th>Default Unit</th>
<th>Value Used For</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone.cpu-shares</td>
<td>cpu-shares</td>
<td>Number of fair share scheduler (FSS) CPU shares for this zone</td>
<td>Quantity (shares)</td>
<td></td>
</tr>
<tr>
<td>zone.max-locked-memory</td>
<td></td>
<td>Total amount of physical locked memory available to a zone.</td>
<td>Size (bytes)</td>
<td>locked property of capped-memory</td>
</tr>
<tr>
<td>zone.max-lwps</td>
<td>max-lwps</td>
<td>Maximum number of LWPs simultaneously available to this zone</td>
<td>Quantity (LWPs)</td>
<td></td>
</tr>
<tr>
<td>zone.max-msg-ids</td>
<td>max-msg-ids</td>
<td>Maximum number of message queue IDs allowed for this zone</td>
<td>Quantity</td>
<td>message queue IDs</td>
</tr>
<tr>
<td>zone.max-sem-ids</td>
<td>max-sem-ids</td>
<td>Maximum number of semaphore IDs allowed for this zone</td>
<td>Quantity</td>
<td>semaphore IDs</td>
</tr>
<tr>
<td>zone.max-shm-ids</td>
<td>max-shm-ids</td>
<td>Maximum number of shared memory IDs allowed for this zone</td>
<td>Quantity</td>
<td>shared memory IDs</td>
</tr>
<tr>
<td>zone.max-shm-memory</td>
<td>max-shm-memory</td>
<td>Total amount of System V shared memory allowed for this zone</td>
<td>Size (bytes)</td>
<td></td>
</tr>
<tr>
<td>zone.max-swap</td>
<td></td>
<td>Total amount of swap that can be consumed by user process address space mappings and tmpfs mounts for this zone.</td>
<td>Size (bytes)</td>
<td>swap property of capped-memory</td>
</tr>
</tbody>
</table>
These limits can be specified for running processes by using the `prctl` command. An example is provided in “How to Set FSS Shares in the Global Zone Using the `prctl` Command” on page 397. Limits specified through the `prctl` command are not persistent. The limits are only in effect until the system is rebooted.

### Configurable Privileges

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

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

**Note** – There are a few privileges that cannot be removed from the zone’s default privilege set, and there are also a few privileges that cannot be added to the set at this time.

For more information, see “Privileges in a Non-Global Zone” on page 367, “How to Configure the Zone” on page 249, and privileges(5).

### Including a Comment for a Zone

You can add a comment for a zone by using the `attr` resource type. For more information, see “How to Configure the Zone” on page 249.

### Using the `zonecfg` Command

The `zonecfg` command, which is described in the `zonecfg(1M)` man page, is used to configure a non-global zone.

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

The `zonecfg` command can be used in interactive mode, in command-line mode, or in command-file mode. The following operations can be performed using this command:

- Create or delete (destroy) a zone configuration
zonecfg Modes

- Add resources to a particular configuration
- Set properties for resources added to a configuration
- Remove resources from a particular configuration
- Query or verify a configuration
- Commit to a configuration
- Revert to a previous configuration
- Rename a zone
- Exit from a zonecfg session

The zonecfg prompt is of the following form:

zonecfg:zonename>

When you are configuring a specific resource type, such as a file system, that resource type is also included in the prompt:

zonecfg:zonename:fs>

For more information, including procedures that show how to use the various zonecfg components described in this chapter, see Chapter 17, “Planning and Configuring Non-Global Zones (Tasks).”

zonecfg Modes

The concept of a scope is used for the user interface. The scope can be either global or resource specific. The default scope is global.

In the global scope, the add subcommand and the select subcommand are used to select a specific resource. The scope then changes to that resource type.

- For the add subcommand, the end or cancel subcommands are used to complete the resource specification.
- For the select subcommand, the end or cancel subcommands are used to complete the resource modification.

The scope then reverts back to global.

Certain subcommands, such as add, remove, and set, have different semantics in each scope.

zonecfg Interactive Mode

In interactive mode, the following subcommands are supported. For detailed information about semantics and options used with the subcommands, see the zonecfg(1M) man page for
options. For any subcommand that could result in destructive actions or loss of work, the system requests user confirmation before proceeding. You can use the -F (force) option to bypass this confirmation.

help  Print general help, or display help about a given resource.

    zonecfg:my-zone:inherit:pkg-dir> help

create  Begin configuring an in-memory configuration for the specified new zone for one of these purposes:

■ To apply the Sun default settings to a new configuration. This method is the default.
■ With the -t template option, to create a configuration that is identical to the specified template. The zone name is changed from the template name to the new zone name.
■ With the -F option, to overwrite an existing configuration.
■ With the -b option, to create a blank configuration in which nothing is set.

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

add  In the global scope, add the specified resource type to the configuration.

In the resource scope, add a property of the given name with the given value.

See "How to Configure the Zone" on page 249 and the zonecfg(1M) man page for more information.

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

select  Applicable only in the global scope. Select the resource of the given type that matches the given property name-property value pair criteria for modification. The scope is changed to that resource type. You must specify a sufficient number of property name-value pairs for the resource to be uniquely identified.

clear  Clear the value for optional settings. Required settings cannot be cleared. However, some required settings can be changed by assigning a new value.

remove  In the global scope, remove the specified resource type. You must specify a sufficient number of property name-value pairs for the resource type to be uniquely identified. If no property name-value pairs are specified, all instances will be removed. If more than one exists, a confirmation is required unless the -F option is used.
In the resource scope, remove the specified property name-property value from the current resource.

end

Applicable only in the resource scope. End the resource specification.

The zonecfg command then verifies that the current resource is fully specified.
- If the resource is fully specified, it is added to the in-memory configuration and the scope will revert back to global.
- If the specification is incomplete, the system displays an error message that describes what needs to be done.

cancel

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

delete

Destroy the specified configuration. Delete the configuration both from memory and from stable storage. You must use the -F (force) option with delete.

Caution – This action is instantaneous. No commit is required, and a deleted zone cannot be reverted.

info

Display information about the current configuration or the global resource properties zonepath, autoboot, and pool. If a resource type is specified, display information only about resources of that type. In the resource scope, this subcommand applies only to the resource being added or modified.

verify

Verify current configuration for correctness. Ensure that all resources have all of their required properties specified.

commit

Commit current configuration from memory to stable storage. Until the in-memory configuration is committed, changes can be removed with the revert subcommand. A configuration must be committed to be used by zoneadm. This operation is attempted automatically when you complete a zonecfg session. Because only a correct configuration can be committed, the commit operation automatically does a verify.

revert

Revert configuration back to the last committed state.

exit

Exit the zonecfg session. You can use the -F (force) option with exit.

A commit is automatically attempted if needed. Note that an EOF character can also be used to exit the session.
zonecfg **Command-File Mode**

In command-file mode, input is taken from a file. The export subcommand described in "zonecfg Interactive Mode" on page 226 is used to produce this file. The configuration can be printed to standard output, or the -f option can be used to specify an output file.

**Zone Configuration Data**

Zone configuration data consists of two kinds of entities: resources and properties. Each resource has a type, and each resource can also have a set of one or more properties. The properties have names and values. The set of properties is dependent on the resource type.

**Resource and Property Types**

The resource and property types are described as follows:

- **zonename**
  
The name of the zone. The following rules apply to zone names:
  
  - Each zone must have a unique name.
  - A zone name is case-sensitive.
  - A zone name must begin with an alphanumeric character.
  
  The name can contain alphanumeric characters, underbars (_), hyphens (-), and periods (.).
  
  - The name cannot be longer than 64 characters.
  - The name global and all names beginning with SUNW are reserved and cannot be used.

- **zonepath**
  
The zonepath property is the path to the zone root. Each zone has a path to its root directory that is relative to the global zone’s root directory. At installation time, the global zone directory is required to have restricted visibility. It must be owned by root with the mode 700.

  The non-global zone’s root path is one level lower. The zone’s root directory has the same ownership and permissions as the root directory (/) in the global zone. The zone directory must be owned by root with the mode 755. These directories are
created automatically with the correct permissions, and do not need to be verified by the zone administrator. This hierarchy ensures that unprivileged users in the global zone are prevented from traversing a non-global zone’s file system.

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home/export/my-zone zonecfg zonepath</td>
<td></td>
</tr>
<tr>
<td>/home/export/my-zone/root</td>
<td>Root of the zone</td>
</tr>
<tr>
<td>/home/export/my-zone/root/Dev</td>
<td>Devices created for the zone</td>
</tr>
</tbody>
</table>

See “Traversing File Systems” on page 358 for a further discussion of this issue.

**Note** – You can move a zone to another location on the same system by specifying a new, full zonepath with the move subcommand of zoneadm. See “Moving a Non-Global Zone” on page 305 for instructions.

**autoboot**

If this property is set to true, the zone is automatically booted when the global zone is booted. Note that if the zones service, svc:/system/zones:default is disabled, the zone will not autoboot, regardless of the setting of this property. You can enable the zones service with the svcadm command described in the svcadm(1M) man page:

```
  global# svcadm enable zones
```

**bootargs**

This property is used to set a boot argument for the zone. The boot argument is applied unless overridden by the reboot, zoneadm boot, or zoneadm reboot commands. See “Zone Boot Arguments” on page 269.

**pool**

This property is used to associate the zone with a resource pool on the system. Multiple zones can share the resources of one pool. Also see “dedicated-cpu Resource” on page 217.
**limitpriv**

This property is used to specify a privilege mask other than the default. See “Privileges in a Non-Global Zone” on page 367.

Privileges are added by specifying the privilege name, with or without the leading `priv_`. Privileges are excluded by preceding the name with a dash (`-`) or an exclamation mark (`!`). The privilege values are separated by commas and placed within quotation marks (`"`).

As described in `priv_str_to_set(3C)`, the special privilege sets of `none`, `all`, and `basic` expand to their normal definitions. Because zone configuration takes place from the global zone, the special privilege set `zone` cannot be used. Because a common use is to alter the default privilege set by adding or removing certain privileges, the special set `default` maps to the default, set of privileges. When `default` appears at the beginning of the `limitpriv` property, it expands to the default set.

The following entry adds the ability to use DTrace programs that only require the `dtrace_proc` and `dtrace_user` privileges in the zone:

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

If the zone's privilege set contains a disallowed privilege, is missing a required privilege, or includes an unknown privilege, an attempt to verify, ready, or boot the zone will fail with an error message.

**scheduling-class**

This property sets the scheduling class for the zone. See “Scheduling Class” on page 218 for additional information and tips.

**ip-type**

This property is required to be set only if the zone is an exclusive-IP zone. See “Exclusive-IP Non-Global Zones” on page 220 and “How to Configure the Zone” on page 249.

**dedicated-cpu**

This resource dedicates a subset of the system's processors to the zone while it is running. The `dedicated-cpu` resource provides limits for `ncpus`
and, optionally, importance. For more information, see "dedicated-cpu Resource" on page 217.

capped-cpu

This resource sets a limit on the amount of CPU resources that can be consumed by the zone while it is running. The capped-cpu resource provides a limit for ncpus. For more information, see "capped-cpu Resource" on page 217.

capped-memory

This resource groups the properties used when capping memory for the zone. The capped-memory resource provides limits for physical, swap, and locked memory. At least one of these properties must be specified.

dataset

Adding a ZFS™ dataset resource enables the delegation of storage administration to a non-global zone. The zone administrator can create and destroy file systems within that dataset, and modify properties of the dataset. The zone administrator cannot affect datasets that have not been added to the zone or exceed any top level quotas set on the dataset assigned to the zone.

ZFS datasets can be added to a zone in the following ways.

- As an lofs mounted file system, when the goal is solely to share space with the global zone
- As a delegated dataset


Also see Chapter 28, “Troubleshooting Miscellaneous Solaris Zones Problems,” for information on dataset issues.

fs

Each zone can have various file systems that are mounted when the zone transitions from the installed state to the ready state. The file system resource specifies the path to the file system mount point. For more information about the use of file systems in zones, see “File Systems and Non-Global Zones” on page 353.
inherit-pkg-dir (native brand only) OpenSolaris native brand: This resource should not be configured in a whole root zone.

In a native branded sparse root zone, the inherit-pkg-dir resource is used to represent directories that contain packaged software that a non-global zone shares with the global zone.

The contents of software packages transferred into the inherit-pkg-dir directory are inherited in read-only mode by the non-global zone. The zone’s packaging database is updated to reflect the packages. These resources cannot be modified or removed after the zone has been installed using zoneadm.

Note – Four default inherit-pkg-dir resources are included in the configuration. These directory resources indicate which directories should have their associated packages inherited from the global zone. The resources are implemented through a read-only loopback file system mount.

- /lib
- /platform
- /sbin
- /usr

net The network interface resource is the interface name. Each zone can have network interfaces that should be set up when the zone transitions from the installed state to the ready state.

device The device resource is the device matching specifier. Each zone can have devices that should be configured when the zone transitions from the installed state to the ready state.

rctl The rctl resource is used for zone-wide resource controls. The controls are enabled when the zone transitions from the installed state to the ready state.

See "Setting Zone-Wide Resource Controls" on page 222 for more information.
Note – To configure zone-wide controls using the set global_property_name subcommand of zonecfg instead of the rctl resource, see “How to Configure the Zone” on page 249.

hostid

A hostid that is different from the hostid of the global zone can be set.

attr

This generic attribute can be used for user comments or by other subsystems. The name property of an attr must begin with an alphanumeric character. The name property can contain alphanumeric characters, hyphens (-), and periods (.). Attribute names beginning with zone. are reserved for use by the system.

Resource Type Properties

Resources also have properties to configure. The following properties are associated with the resource types shown.

dedicated-cpu ncpus, importance

Specify the number of CPUs and, optionally, the relative importance of the pool. The following example specifies a CPU range for use by the zone my-zone. importance is also set.

```
zonecfg:my-zone> add dedicated-cpu
zonecfg:my-zone:dedicated-cpu> set ncpus=1-3
zonecfg:my-zone:dedicated-cpu> set importance=2
zonecfg:my-zone:dedicated-cpu> end
```

capped-cpu ncpus

Specify the number of CPUs. The following example specifies a CPU cap of 3.5 CPUs for the zone my-zone.

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

capped-memory physical, swap, locked

Specify the memory limits for the zone my-zone. Each limit is optional, but at least one must be set.
The `fs` resource parameters supply the values that determine how and where to mount file systems. The `fs` parameters are defined as follows:

- **dir**: Specifies the mount point for the file system.
- **special**: Specifies the block special device name or directory from the global zone to mount.
- **raw**: Specifies the raw device on which to run `fsck` before mounting the file system.
- **type**: Specifies the file system type.
- **options**: Specifies mount options similar to those found with the `mount` command.

The lines in the following example specify that `/dev/dsk/c0t0d0s2` in the global zone is to be mounted as `/mnt` in a zone being configured. The `raw` property specifies an optional device on which the `fsck` command is to be run before an attempt is made to mount the file system. The file system type to use is `UFS`. The options `nodoses` and `logging` are added.

For more information, see “The `-o nosuid` Option” on page 353, “Security Restrictions and File System Behavior” on page 356, and the `fsck(1M)` and `mount(1M)` man pages. Also note that section 1M man pages are available for mount options that are unique to a specific file system. The names of these man pages have the form `mount_filesystem`. 
Note – To add a ZFS file system using the fs resource property, see “Adding ZFS File Systems to a Non-Global Zone” in Solaris ZFS Administration Guide.

**dataset**

The lines in the following example specify that the dataset sales is to be visible and mounted in the non-global zone and no longer visible in the global zone.

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

**inherit-pkg-dir**

The lines in the following example specify that /opt/sfw is to be loopback mounted from the global zone.

```
zonecfg:my-zone> add inherit-pkg-dir
zonecfg:my-zone:inherit-pkg-dir> set dir=/opt/sfw
zonecfg:my-zone:inherit-pkg-dir> end
```

**net**

Note – For a shared-IP zone, both the IP address and the device are specified. Optionally, the default router can be set. For an exclusive-IP zone, only the physical interface is specified.

In the following example for a shared-IP zone, the IP address 192.168.0.1 is added to the zone. An hme0 card is used for the physical interface. To determine which physical interface to use, type `ifconfig` -a on your system. Each line of the output, other than loopback driver lines, begins with the name of a card installed on your system. Lines that contain LOOPBACK in the descriptions do not apply to cards. The default route is set to 10.0.0.1 for the zone.

```
zonectf:my-zone> add net
zonectf:my-zone:net> set physical=hme0
zonectf:my-zone:net> set address=192.168.0.1
zonectf:my-zone:net> set defrouter=10.0.0.1
zonectf:my-zone:net> end
```
In the following example for an exclusive-IP zone, a bge32001 link is used for the physical interface, which is a VLAN on bge1. To determine which data-links are available, use the command `dladm show-link`. Note that `ip-type=exclusive` must also be specified.

```
zonecfg:my-zone> set ip-type=exclusive
zonecfg:my-zone> add net
zonecfg:my-zone:net> set physical=bge32001
zonecfg:my-zone:net> end
```

**Note** – The OpenSolaris™ OS supports all Ethernet-type interfaces, and their data-links can be administered with the `dladm` command. Prior to OpenSolaris build snv_83, the data-link must be GLDv3 to be used with exclusive-IP zones. Network interface cards (NICs) that support GLDv3 include bge, e1000g, xge, rge, and rge. The ce legacy device can also support an exclusive-IP zone.

```
device match
```

In the following example, a `/dev/pts` device is included in a zone.

```
zonecfg:my-zone> add device
zonecfg:my-zone:device> set match=/dev/pts*
zonecfg:my-zone:device> end
```

**Note** – See “Device Use in Non-Global Zones” on page 364.

```
rctl name, value
```

The following zone-wide resource controls are available.

- `zone.cpu-cap`
- `zone.cpu-shares` (preferred: `cpu-shares`)
- `zone.max-locked-memory`
- `zone.max-lwps` (preferred: `max-lwps`)
- `zone.max-msg-ids` (preferred: `max-msg-ids`)
- `zone.max-sem-ids` (preferred: `max-sem-ids`)
- `zone.max-shm-ids` (preferred: `max-shm-ids`)
- `zone.max-shm-memory` (preferred: `max-shm-memory`)
- `zone.max-swap`

Note that the preferred, simpler method for setting a zone-wide resource control is to use the property name instead of the `rctl` resource, as
shown in "How to Configure the Zone" on page 249. If zone-wide resource control entries in a zone are configured using add rctl, the format is different than resource control entries in the project database. In a zone configuration, the rctl resource type consists of three name/value pairs. The names are priv, limit, and action. Each of the names takes a simple value.

```bash
zonecfg:my-zone> add rctl
zonecfg:my-zone:rctl> set name=zone.cpu-shares
zonecfg:my-zone:rctl> add value (priv=privileged,limit=10,action=none)
zonecfg:my-zone:rctl> end
```

```bash
zonecfg:my-zone> add rctl
zonecfg:my-zone:rctl> set name=zone.max-lwps
zonecfg:my-zone:rctl> add value (priv=privileged,limit=100,action=deny)
zonecfg:my-zone:rctl> end
```

For general information about resource controls and attributes, see Chapter 6, "Resource Controls (Overview)," and "Resource Controls Used in Non-Global Zones" on page 366.

```bash
attr name, type, value
```

In the following example, a comment about a zone is added.

```bash
zonecfg:my-zone> add attr
zonecfg:my-zone:attr> set name=comment
zonecfg:my-zone:attr> set type=string
zonecfg:my-zone:attr> set values="Production zone"
zonecfg:my-zone:attr> end
```

You can use the export subcommand to print a zone configuration to standard output. The configuration is saved in a form that can be used in a command file.

---

**Teclacommand-Line Editing Library**

The Teclacommand-line editing library is included for use with the zonecfg command. The library provides a mechanism for command-line history and editing support.

The Teclacommand-line editing library is documented in the following man pages:

- enhance(1)
- libtecla(3LIB)
- ef_expand_file(3TECLA)
- gl_get_line(3TECLA)
- gl_io_mode(3TECLA)
- pca_lookup_file(3TECLA)
- tecla(5)
This chapter describes what you need to do before you can configure a zone on your system. This chapter also describes how to configure a zone, modify a zone configuration, and delete a zone configuration from your system.

For an introduction to the zone configuration process, see Chapter 16, "Non-Global Zone Configuration (Overview)."

For information about \texttt{lx} branded zone configuration, see Chapter 30, "Planning the \texttt{lx} Branded Zone Configuration (Overview)," and Chapter 31, "Configuring the \texttt{lx} Branded Zone (Tasks)."

**Planning and Configuring a Non-Global Zone (Task Map)**

Before you set up your system to use zones, you must first collect information and make decisions about how to configure the zones. The following task map summarizes how to plan and configure a zone.
<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
</table>
| Plan your zone strategy. | ▪ Evaluate the applications running on your system to determine which applications you want to run in a zone.  
▪ Assess the availability of disk space to hold the files that are unique in the zone.  
▪ If you are also using resource management features, determine how to align the zone with the resource management boundaries.  
▪ If you are using resource pools, configure the pools if necessary. | Refer to historical usage. Also see “Disk Space Requirements” on page 244 and “Resource Pools Used in Zones” on page 145. |
| Determine the name for the zone. | Decide what to call the zone based on the naming conventions. | See “Zone Configuration Data” on page 229 and “Zone Host Name” on page 246. |
| Determine the zone path. | Each zone has a path to its root directory that is relative to the global zone's root directory. | See “Zone Configuration Data” on page 229. |
| Evaluate the need for CPU restriction if you are not configuring resource pools. Note that with specification in zonecfg, pool settings propagate during migrations. | Review your application requirements. | See “dedicated-cpu Resource” on page 217. |
| Evaluate the need for memory allocation if you plan to cap memory for the zone by using rcapd from the global zone. | Review your application requirements. | See Chapter 10, “Physical Memory Control Using the Resource Capping Daemon (Overview),”  
<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make the FSS the default scheduler on the system.</td>
<td>Give each zone CPU shares to control the zone’s entitlement to CPU resources. The FSS guarantees a fair dispersion of CPU resources among zones that is based on allocated shares.</td>
<td>Chapter 8, “Fair Share Scheduler (Overview),” “Scheduling Class” on page 218.</td>
</tr>
<tr>
<td>Determine whether the zone will be a shared-IP zone or an exclusive-IP zone.</td>
<td>For a shared-IP zone, which is the default, obtain or configure IP addresses for the zone. Depending on your configuration, you must obtain at least one IP address for each non-global zone that you want to have network access. For an exclusive-IP zone, determine the data-link that will be assigned to the zone. The zone requires exclusive access to one or more network interfaces. The interface could be a separate LAN such as bge1, or a separate VLAN such as bge2000. Prior to OpenSolaris™ build 83, the data-link must be GLDv3. The ce legacy device is an exception. This device can support an exclusive-IP zone.</td>
<td>See “Determine the Zone Host Name and the Network Requirements” on page 246, “How to Configure the Zone” on page 249, and System Administration Guide: IP Services.</td>
</tr>
<tr>
<td>Determine which file systems you want to mount in the zone.</td>
<td>Review your application requirements.</td>
<td>See “File Systems Mounted in Zones” on page 221 for more information.</td>
</tr>
<tr>
<td>Determine which network interfaces should be made available in the zone.</td>
<td>Review your application requirements.</td>
<td>See “Shared-IP Network Interfaces” on page 361 for more information.</td>
</tr>
<tr>
<td>Determine whether you must alter the default set of non-global zone permissions.</td>
<td>Check the set of privileges: default, privileges that can be added and removed, and privileges that cannot be used at this time.</td>
<td>See ”Privileges in a Non-Global Zone” on page 367.</td>
</tr>
<tr>
<td>Determine which devices should be configured in each zone.</td>
<td>Review your application requirements.</td>
<td>Refer to the documentation for your application.</td>
</tr>
<tr>
<td>Configure the zone.</td>
<td>Use zonecfg to create a configuration for the zone.</td>
<td>See ”Configuring, Verifying, and Committing a Zone” on page 249.</td>
</tr>
</tbody>
</table>
Evaluating the Current System Setup

Zones can be used on any machine that runs the Solaris 10 or later release. The following primary machine considerations are associated with the use of zones.

- The performance requirements of the applications running within each zone.
- The availability of disk space to hold the files that are unique within each zone.

Disk Space Requirements

There are no limits on how much disk space can be consumed by a zone. The global administrator is responsible for space restriction. The global administrator must ensure that local storage is sufficient to hold a non-global zone’s root file system. Even a small uniprocessor system can support a number of zones running simultaneously.

The nature of the packages installed in the global zone affects the space requirements of the non-global zones that are created. The number of packages and space requirements are factors.

Sparse Root Zones

Non-global zones that have `inherit-pkg-dir` resources are called sparse root zones. On OpenSolaris 2006.09, sparse root zones are native branded zones. See the `native(5)` man page for more information on these types of branded zones.

The sparse root zone model optimizes the sharing of objects in the following ways:

- Only a subset of the packages installed in the global zone are installed directly into the non-global zone.
- Read-only loopback file systems, identified as `inherit-pkg-dir` resources, are used to gain access to other files.

In this model, all packages appear to be installed in the non-global zone. Packages that do not deliver content into read-only loopback mount file systems are fully installed. There is no need to install content delivered into read-only loopback mounted file systems since that content is inherited (and visible) from the global zone.

- As a general guideline, a zone requires about 100 megabytes of free disk space per zone when the global zone has been installed with all of the standard Solaris packages.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify and commit the configured zone.</td>
<td>Determine whether the resources and properties specified are valid on a hypothetical system.</td>
<td>See “Configuring, Verifying, and Committing a Zone” on page 249.</td>
</tr>
</tbody>
</table>
By default, any additional packages installed in the global zone also populate the non-global zones. The amount of disk space required might be increased accordingly, depending on whether the additional packages deliver files that reside in the `inherit-pkg-dir` resource space.

An additional 40 megabytes of RAM per zone are suggested, but not required on a machine with sufficient swap space.

**Whole Root Zones**

The whole root zone model provides the maximum configurability. All of the required and any selected optional Solaris packages are installed into the private file systems of the zone. The advantages of this model include the capability for global administrators to customize their zones file system layout. This would be done, for example, to add arbitrary unbundled or third-party packages.

The disk requirements for this model are determined by the disk space used by the packages currently installed in the global zone.

---

**Note** – If you create a sparse root zone that contains the following `inherit-pkg-dir` directories, you must remove these directories from the non-global zone’s configuration before the zone is installed to have a whole root zone:

- `/lib`
- `/platform`
- `/sbin`
- `/usr`

See “How to Configure the Zone” on page 249.

---

**Restricting Zone Size**

The following options can be used to restrict zone size:

- You can place the zone on a `lofi`-mounted partition. This action will limit the amount of space consumed by the zone to that of the file used by `lofi`. For more information, see the `lofiadm(1M)` and `lofi(7D)` man pages.

- You can use soft partitions to divide disk slices or logical volumes into partitions. You can use these partitions as zone roots, and thus limit per-zone disk consumption. The soft partition limit is 8192 partitions. For more information, see Chapter 12, “Soft Partitions (Overview),” in *Solaris Volume Manager Administration Guide*.

- You can use the standard partitions of a disk for zone roots, and thus limit per-zone disk consumption.
Determine the Zone Host Name and the Network Requirements

You must determine the host name for the zone. Then, for a shared IP zone, you must assign an IPv4 address or manually configure and assign an IPv6 address for the zone if you want it to have network connectivity. Inside an exclusive-IP zone, you configure addresses as you do for the global zone.

Zone Host Name

The hostname you select for the zone must be defined either in the hosts database or in the /etc/inet/hosts database, as specified by the /etc/nsswitch.conf file in the global zone. The network databases are files that provide network configuration information. The nsswitch.conf file specifies which naming service to use.

If you use local files for the naming service, the hosts database is maintained in the /etc/inet/hosts file. The host names for zone network interfaces are resolved from the local hosts database in /etc/inet/hosts. Alternatively, the IP address itself can be specified directly when configuring a zone so that no host name resolution is required.


Shared-IP Zone Network Address

Each shared-IP zone that requires network connectivity has one or more unique IP addresses. Both IPv4 and IPv6 addresses are supported.

IPv4 Zone Network Address

If you are using IPv4, obtain an address and assign the address to the zone.

A prefix length can also be specified with the IP address. The format of this prefix is address/prefix-length, for example, 192.168.1.1/24. Thus, the address to use is 192.168.1.1 and the netmask to use is 255.255.255.0, or the mask where the first 24 bits are 1-bits.

IPv6 Zone Network Address

If you are using IPv6, you must manually configure the address. Typically, at least the following two types of addresses must be configured:

Link-local address
A link-local address is of the form fe80::64-bit interface ID/10. The /10 indicates a prefix length of 10 bits.
Address formed from a global prefix configured on the subnet
A global unicast address is based off a 64–bit prefix that the administrator configures for each subnet, and a 64-bit interface ID. The prefix can also be obtained by running the \texttt{ifconfig} command with the \texttt{-a6} option on any system on the same subnet that has been configured to use IPv6.

The 64–bit interface ID is typically derived from a system’s MAC address. For zones use, an alternate address that is unique can be derived from the global zone’s IPv4 address as follows:

16 bits of zero:upper 16 bits of IPv4 address:lower 16 bits of IPv4 address: a zone-unique number

For example, if the global zone’s IPv4 address is 192.168.200.10, a suitable link-local address for a non-global zone using a zone-unique number of 1 is \texttt{fe80::c0a8:c80a:1/10}. If the global prefix in use on that subnet is \texttt{2001:0db8:aabb:ccdd/64}, a unique global unicast address for the same non-global zone is \texttt{2001:0db8:aabb:ccdd::c0a8:c80a:1/64}. Note that you must specify a prefix length when configuring an IPv6 address.

For more information about link-local and global unicast addresses, see the \texttt{inet6(7P)} man page.

**Exclusive-IP Zone Network Address**

Inside an exclusive-IP zone, configure addresses as you do for the global zone. Note that DHCP and IPv6 stateless address autoconfiguration can be used to configure addresses.

See \texttt{sysidcfg(4)} for more information.

**File System Configuration**

You can specify a number of mounts to be performed when the virtual platform is set up. File systems that are loopback-mounted into a zone by using the loopback virtual file system (LOFS) file system should be mounted with the \texttt{nodevices} option. For information on the \texttt{nodevices} option, see “File Systems and Non-Global Zones” on page 353.

LOFS lets you create a new virtual file system so that you can access files by using an alternative path name. In a non-global zone, a loopback mount makes the file system hierarchy look as though it is duplicated under the zone’s root. In the zone, all files will be accessible with a path name that starts from the zone’s root. LOFS mounting preserves the file system name space.
Creating, Revising, and Deleting Non-Global Zone Configurations (Task Map)

See the `lofs(7S)` man page for more information.

### FIGURE 17–1  Loopback-Mounted File Systems

Creating, Revising, and Deleting Non-Global Zone Configurations (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure a non-global zone.</td>
<td>Use the <code>zonecfg</code> command to create a zone, verify the configuration, and commit the configuration. You can also use a script to configure and boot multiple zones on your system. You can use the <code>zonecfg</code> command to display the configuration of a non-global zone.</td>
<td>“Configuring, Verifying, and Committing a Zone” on page 249, “Script to Configure Multiple Zones” on page 255</td>
</tr>
</tbody>
</table>
Configuring, Verifying, and Committing a Zone

You use the zonecfg command described in the zonecfg(1M) man page to perform the following actions.

- Create the zone configuration
- Verify that all required information is present
- Commit the non-global zone configuration

The zonecfg command can also be used to persistently specify the resource management settings for the global zone.

While configuring a zone with the zonecfg utility, you can use the revert subcommand to undo the setting for a resource. See "How to Revert a Zone Configuration" on page 261.

A script to configure multiple zones on your system is provided in “Script to Configure Multiple Zones” on page 255.

To display a non-global zone’s configuration, see “How to Display the Configuration of a Non-Global Zone” on page 257.

▼ How to Configure the Zone

Note that the only required elements to create a native non-global zone are the zonename and zonepath properties. Other resources and properties are optional. Some optional resources also require choices between alternatives, such as the decision to use either the dedicated-cpu resource or the capped-cpu resource. See “Zone Configuration Data” on page 229 for information on available zonecfg properties and resources.
You must be the global administrator in the global zone to perform this procedure.

1 **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration.*

2 **Set up a zone configuration with the zone name you have chosen.**
   The name *my-zone* is used in this example procedure.

   ```
   global# zonecfg -z my-zone
   ```
   If this is the first time you have configured this zone, you will see the following system message:
   ```
   my-zone: No such zone configured
   Use 'create' to begin configuring a new zone.
   ```

3 **Create the new zone configuration.**
   This procedure uses the Sun default settings.

   ```
   zonecfg:my-zone> create
   ```

4 **Set the zone path, */export/home/my-zone in this procedure.**

   ```
   zonecfg:my-zone> set zonepath=/export/home/my-zone
   ```

5 **Set the autoboot value.**
   If set to *true*, the zone is automatically booted when the global zone is booted. Note that for the zones to autoboott, the zones service `svc:/system/zones:default` must also be enabled. The default value is *false*.

   ```
   zonecfg:my-zone> set autoboot=true
   ```

6 **Set persistent boot arguments for a zone.**

   ```
   zonecfg:my-zone> set bootargs="-m verbose"
   ```

7 **Dedicate one CPU to this zone.**

   ```
   zonecfg:my-zone> add dedicated-cpu
   ```
   a. **Set the number of CPUs.**

      ```
      zonecfg:my-zone:dedicated-cpu> set ncpus=1-2
      ```
   b. **(Optional) Set the importance.**

      ```
      zonecfg:my-zone:dedicated-cpu> set importance=10
      ```
c. End the specification.
   zonecfg:my-zone:dedicated-cpu> end

8 Revise the default set of privileges.
   zonecfg:my-zone> set limitpriv="default, sys_time"
   This line adds the ability to set the system clock to the default set of privileges.

9 Set the scheduling class to FSS.
   zonecfg:my-zone> set scheduling-class=FSS

10 Add a memory cap.
   zonecfg:my-zone> add capped-memory
   a. Set the memory cap.
      zonecfg:my-zone:capped-memory> set physical=50m
   b. Set the swap memory cap.
      zonecfg:my-zone:capped-memory> set swap=100m
   c. Set the locked memory cap.
      zonecfg:my-zone:capped-memory> set locked=30m
   d. End the memory cap specification.
      zonecfg:my-zone:capped-memory> end

11 Add a file system.
   zonecfg:my-zone> add fs
   a. Set the mount point for the file system, /usr/local in this procedure.
      zonecfg:my-zone:fs> set dir=/usr/local
   b. Specify that /opt/local in the global zone is to be mounted as /usr/local in the zone being configured.
      zonecfg:my-zone:fs> set special=/opt/local
      In the non-global zone, the /usr/local file system will be readable and writable.
   c. Specify the file system type, lofs in this procedure.
      zonecfg:my-zone:fs> set type=lofs
      The type indicates how the kernel interacts with the file system.
d. End the file system specification.

   zonecfg:my-zone:fs> end

   This step can be performed more than once to add more than one file system.

12  (Optional) Set the hostid.

   zonecfg:my-zone> set hostid=80f0c086

13  Add a ZFS dataset named sales in the storage pool tank

   zonecfg:my-zone> add dataset

   a. Specify the path to the ZFS dataset sales.

      zonecfg:my-zone> set name=tank/sales

   b. End the dataset specification.

      zonecfg:my-zone> end

14  (Sparse Root Zone Only) Add a shared filesystem that is loopback-mounted from the global zone.

   Do not perform this step to create a whole root zone, which does not have any shared file systems. See the discussion for whole root zones in "Disk Space Requirements" on page 244.

   zonecfg:my-zone> add inherit-pkg-dir

   a. Specify that /opt/sfw in the global zone is to be mounted in read-only mode in the zone being configured.

      zonecfg:my-zone:inherit-pkg-dir> set dir=/opt/sfw

      Note – The zone's packaging database is updated to reflect the packages. These resources cannot be modified or removed after the zone has been installed using zoneadm.

   b. End the inherit-pkg-dir specification.

      zonecfg:my-zone:inherit-pkg-dir> end

   This step can be performed more than once to add more than one shared file system.
**Note** – If you want to create a whole root zone but default shared file systems resources have been added by using *inherit-pkg-dir*, you must remove these default *inherit-pkg-dir* resources using `zonecfg` before you install the zone:

- `zonecfg:my-zone> remove inherit-pkg-dir dir=/lib`
- `zonecfg:my-zone> remove inherit-pkg-dir dir=/platform`
- `zonecfg:my-zone> remove inherit-pkg-dir dir=/sbin`
- `zonecfg:my-zone> remove inherit-pkg-dir dir=/usr`

15 **(Optional) If you are creating an exclusive-IP zone, set the** `ip-type`.
   
   `zonecfg:my-zone> set ip-type=exclusive`

   **Note** – Only the physical device type will be specified in the add net step.

16 **Add a network interface.**

   `zonecfg:my-zone> add net`

   a. **(shared-IP only) Set the IP address for the network interface, 192.168.0.1 in this procedure.**

   `zonecfg:my-zone:net> set address=192.168.0.1`

   b. **Set the physical device type for the network interface, the hme device in this procedure.**

   `zonecfg:my-zone:net> set physical=hme0`

   c. **(Optional, shared-IP only) Set the default router for the network interface, in this procedure.**

   `zonecfg:my-zone:net> set defrouter=10.0.0.1`

   d. **End the specification.**

   `zonecfg:my-zone:net> end`

   This step can be performed more than once to add more than one network interface.

17 **Add a device.**

   `zonecfg:my-zone> add device`

   a. **Set the device match, */dev/sound/* in this procedure.**

   `zonecfg:my-zone:device> set match=/dev/sound/*`

   b. **End the device specification.**

   `zonecfg:my-zone:device> end`

   This step can be performed more than once to add more than one device.
Add a zone-wide resource control by using the property name.

```
zonecfg:my-zone> set max-sem-ids=10485200
```

This step can be performed more than once to add more than one resource control.

Add a comment by using the `attr` resource type.

```
zonecfg:my-zone> add attr
```

a. Set the name to `comment`.
```
zonecfg:my-zone:attr> set name=comment
```

b. Set the type to `string`.
```
zonecfg:my-zone:attr> set type=string
```

c. Set the value to a comment that describes the zone.
```
zonecfg:my-zone:attr> set value="This is my work zone."
```

d. End the `attr` resource type specification.
```
zonecfg:my-zone:attr> end
```

Verify the zone configuration for the zone.
```
zonecfg:my-zone> verify
```

Commit the zone configuration for the zone.
```
zonecfg:my-zone> commit
```

Exit the `zonecfg` command.
```
zonecfg:my-zone> exit
```

Note that even if you did not explicitly type `commit` at the prompt, a commit is automatically attempted when you type `exit` or an EOF occurs.

Using Multiple Subcommands From the Command Line

**Tip** – The `zonecfg` command also supports multiple subcommands, quoted and separated by semicolons, from the same shell invocation.

```
global# zonecfg -z my-zone "create ; set zonepath=/export/home/my-zone"
```

Where to Go From Here

See "Installing and Booting Zones" on page 274 to install your committed zone configuration.
**Script to Configure Multiple Zones**

You can use this script to configure and boot multiple zones on your system. The script takes the following parameters:

- The number of zones to be created
- The `zonename` prefix
- The directory to use as the base directory

You must be the global administrator in the global zone to execute the script. The global administrator has superuser privileges in the global zone or assumes the Primary Administrator role.

```bash
#!/bin/ksh
#
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
ident "%Z%M %I% %E% SMI"

if [[ -z "$1" || -z "$2" || -z "$3" ]]; then
    echo "usage: $0 <# of zones> <zonename-prefix> <basedir>"
    exit 2
fi

if [[ ! -d $3 ]]; then
    echo "$3 is not a directory"
    exit 1
fi

nprocs=`psrinfo | wc -l`
nzones=$1
prefix=$2
dir=$3

ip_addrs_per_if='ndd /dev/ip ip_addrs_per_if'
if [ $ip_addrs_per_if -lt $nzones ]; then
    echo "ndd parameter ip_addrs_per_if is too low ($ip_addrs_per_if)"
    echo "set it higher with 'ndd -set /dev/ip ip_addrs_per_if <num>"
    exit 1
fi

i=1
while [ $i -le $nzones ]; do
    zoneadm -z $prefix$i list > /dev/null 2>&1
    if [ $? != 0 ]; then
        echo configuring $prefix$i
        F=$dir/$prefix$i.config
        Configuring, Verifying, and Committing a Zone
```

Chapter 17 • Planning and Configuring Non-Global Zones (Tasks) 255
Configuring, Verifying, and Committing a Zone

```
rm -f $F
echo "create" > $F
echo "set zonepath=${dir}/${prefix$i}" >> $F
zonecfg -z ${prefix$i} -f ${dir}/${prefix$i}.config 2>&1 | \
   sed 's/\'/ /g'
else
echo "skipping ${prefix$i}, already configured"
fi
```

```
i='expr $i + 1'
done
```

```
i=1
while [ $i -le $nzones ]; do
  j=1
  while [ $j -le $nprocs ]; do
    if [ $i -le $nzones ]; then
      if [ "zoneadm -z ${prefix$i} list -p | \ 
           cut -d:'' -f 3'" != "configured" ]; then
        echo "skipping ${prefix$i}, already installed"
      else
        echo installing ${prefix$i}
        mkdir -pm 0700 ${dir}/${prefix$i}
        chmod 700 ${dir}/${prefix$i}
        zoneadm -z ${prefix$i} install > /dev/null 2>&1 &
        sleep 1 # spread things out just a tad
      fi
    fi
    i='expr $i + 1'
    j='expr $j + 1'
  done
  wait
done
```

```
i=1
while [ $i -le $nzones ]; do
  echo setting up sysid for ${prefix$i}
  cfg=${dir}/${prefix$i}/root/etc/sysidcfg
  rm -f $cfg
  echo "network_interface=NONE {hostname=${prefix$i}}" > $cfg
  echo "system.locale=C" >> $cfg
  echo "terminal=xterms" >> $cfg
  echo "security_policy=NONE" >> $cfg
  echo "name_service=NONE" >> $cfg
  echo "timezone=US/Pacific" >> $cfg
  echo "root_password=Qexr7Y/wzk5bc" >> $cfg # 'l1a'
  i='expr $i + 1'
done
```
i=1
para=`expr $nprocs \* 2`
while [ $i -le $nzones ]; do
date
j=1
while [ $j -le $para ]; do
  if [ $i -le $nzones ]; then
    echo booting $prefix$i
    zoneadm -z $prefix$i boot &
  fi
  j=`expr $j + 1`
  i=`expr $i + 1`
done
wait
done

▼ How to Display the Configuration of a Non-Global Zone

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Display the configuration of a zone.
   global# zonectf -z zonename info

Using the zonectf Command to Modify a Zone Configuration

You can also use the zonectf command to do the following:

- Modify a resource type in a zone configuration
- Clear a property value in a zone configuration
- Add a dedicated device to a zone

▼ How to Modify a Resource Type in a Zone Configuration

You can select a resource type and modify the specification for that resource.

Note that the contents of software packages in the inherit-pkg-dir directory cannot be modified or removed after the zone has been installed with zoneadm.
You must be the global administrator in the global zone to perform this procedure.

1 **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2 **Select the zone to be modified, my-zone in this procedure.**
   ```
   global# zonecfg -z my-zone
   ```

3 **Select the resource type to be changed, for example, a resource control.**
   ```
   zonecfg:my-zone> select rctl name=zone.cpu-shares
   ```

4 **Remove the current value.**
   ```
   zonecfg:my-zone:rctl> remove value (priv=privileged,limit=20,action=none)
   ```

5 **Add the new value.**
   ```
   zonecfg:my-zone:rctl> add value (priv=privileged,limit=10,action=none)
   ```

6 **End the revised rctl specification.**
   ```
   zonecfg:my-zone:rctl> end
   ```

7 **Commit the zone configuration for the zone.**
   ```
   zonecfg:my-zone> commit
   ```

8 **Exit the zonecfg command.**
   ```
   zonecfg:my-zone> exit
   ```

   Note that even if you did not explicitly type `commit` at the prompt, a `commit` is automatically attempted when you type `exit` or an EOF occurs.

   Committed changes made through `zonecfg` take effect the next time the zone is booted.

**How to Clear a Property Type in a Zone Configuration**

Use this procedure to reset a standalone property.

1 **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2 **Select the zone to be modified, my-zone in this procedure.**
   ```
   global# zonecfg -z my-zone
   ```
3 Clear the property to be changed, the existing pool association in this procedure.
   zonecfg:my-zone> clear pool

4 Commit the zone configuration for the zone.
   zonecfg:my-zone> commit

5 Exit the zonecfg command.
   zonecfg:my-zone> exit

   Note that even if you did not explicitly type commit at the prompt, a commit is automatically attempted when you type exit or an EOF occurs.

   Committed changes made through zonecfg take effect the next time the zone is booted.

▼ How to Rename a Zone

This procedure can be used to rename zones that are in either the configured state or the installed state.

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Select the zone to be renamed, my-zone in this procedure.
   global# zonecfg -z my-zone

3 Change the name of the zone, for example, to newzone.
   zonecfg:my-zone> set zonename=newzone

4 Commit the change.
   zonecfg:newzone> commit

5 Exit the zonecfg command.
   zonecfg:newzone> exit

   Committed changes made through zonecfg take effect the next time the zone is booted.
How to Add a Dedicated Device to a Zone

The following specification places a scanning device in a non-global zone configuration.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Add a device.**
   ```
   zonecfg:my-zone> add device
   ```

3. **Set the device match, /dev/scsi/scanner/c3t4* in this procedure.**
   ```
   zonecfg:my-zone:device> set match=/dev/scsi/scanner/c3t4*
   ```

4. **End the device specification.**
   ```
   zonecfg:my-zone:device> end
   ```

5. **Exit the zonecfg command.**
   ```
   zonecfg:my-zone> exit
   ```

How to Set zone.cpu-shares in the Global Zone

This procedure is used to persistently set shares in the global zone.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Use the zonecfg command.**
   ```
   # zonecfg -z global
   ```

3. **Set five shares for the global zone.**
   ```
   zonecfg:global> set cpu-shares=5
   ```

4. **Exit zonecfg.**
   ```
   zonecfg:global> exit
   ```
Using the `zonecfg` Command to Revert or Remove a Zone Configuration

Use the `zonecfg` command described in `zonecfg(1M)` to revert a zone's configuration or to delete a zone configuration.

▼ How to Revert a Zone Configuration

While configuring a zone with the `zonecfg` utility, use the `revert` subcommand to undo a resource setting made to the zone configuration.

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 While configuring a zone called `tmp-zone`, type `info` to view your configuration:
   
   ```
   zonecfg:tmp-zone> info
   ```
   The net resource segment of the configuration displays as follows:

   ```
   .
   .
   .
   fs:
     dir: /tmp
     special: swap
     type: tmpfs
   net:
     address: 192.168.0.1
     physical: eri0
   device
     match: /dev/pts/*
   .
   .
   .
   ```

3 Remove the net address:
   
   ```
   zonecfg:tmp-zone> remove net address=192.168.0.1
   ```
4 Verify that the net entry has been removed.
zonecfg:tmp-zone> info

.
.
 fs:
     dir: /tmp
     special: swap
     type: tmpfs
device
     match: /dev/pts/*
.
.
5 Type revert.
zonecfg:tmp-zone> revert

6 Answer yes to the following question:
Are you sure you want to revert (y/[n])? y

7 Verify that the net address is once again present:
zonecfg:tmp-zone> info

.
.
.
 fs:
     dir: /tmp
     special: swap
     type: tmpfs
 net:
     address: 192.168.0.1
     physical: eri0
device
     match: /dev/pts/*
.
.

How to Delete a Zone Configuration

Use `zonecfg` with the `delete` subcommand to delete a zone configuration from the system.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2. **Delete the zone configuration for the zone `a-zone` by using one of the following two methods:**
   
   - Use the `-F` option to force the action:
     
     ```
     global# zonecfg -z a-zone delete -F
     ```
   
   - Delete the zone interactively by answering yes to the system prompt:
     
     ```
     global# zonecfg -z a-zone delete
     Are you sure you want to delete zone a-zone (y/[n])? y
     ```
This chapter discusses zone installation on your Solaris system. It also describes the two processes that manage the virtual platform and the application environment, zoneadm and zsched. Information about halting, rebooting, cloning, and uninstalling zones is also provided.

The following topics are addressed in this chapter:

- “Zone Installation and Administration Concepts” on page 265
- “Zone Construction” on page 266
- “The zoneadm Daemon” on page 268
- “The zsched Zone Scheduler” on page 268
- “Zone Application Environment” on page 268
- “About Halting, Rebooting, and Uninstalling Zones” on page 269
- “About Cloning Non-Global Zones” on page 271

To clone a non-global zone, install and boot a non-global zone, or to halt or uninstall a non-global zone, see Chapter 19, “Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks).”

For information about 1x branded zone installation, see Chapter 32, “About Installing, Booting, Halting, Cloning, and Uninstalling 1x Branded Zones (Overview),” and Chapter 33, “Installing, Booting, Halting, Uninstalling and Cloning 1x Branded Zones (Tasks).”

Zone Installation and Administration Concepts

The zoneadm command described in the zoneadm(1M) man page is the primary tool used to install and administer non-global zones. Operations using the zoneadm command must be run from the global zone. The following tasks can be performed using the zoneadm command:

- Verify a zone
- Install a zone
- Change the state of an installed zone to incomplete
Zone Construction

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

After you have configured a non-global zone, you should verify that the zone can be installed safely on your system's configuration. You can then install the zone. The files needed for the zone's root file system are installed by the system under the zone's root path.

A non-global zone is installed with the limited networking configuration (generic_limited_net.xml). Network configuration types are described in Chapter 17, "Managing Services (Tasks)," in System Administration Guide: Basic Administration. The zone administrator can switch the zone to the open, traditional networking configuration (generic_open.xml) by using the netservices command. Specific services can be enabled or disabled by using SMF commands. For more information, see "Switching the Non-Global Zone to a Different Networking Service Configuration" on page 301.

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

The method used to initially install packages in a Solaris installation is also the method used to populate a non-global zone.

The global zone must contain all the data necessary to populate a non-global zone. Populating a zone includes creating directories, copying files, and providing configuration information.
Only the information or data that was created in the global zone from packages is used to populate the zone from the global zone. For more information, see the `pkgparam(1)` and `pkginfo(4)` man pages.

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

- Non-installed packages
- Patches
- Data on CDs and DVDs
- Network installation images
- Any prototype or other instance of a zone

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

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

If Solaris auditing is used, modifications to auditing files copied from the global zone might be required. For more information, see “Using Solaris Auditing in Zones” on page 372.

The following features cannot be configured in a non-global zone:

- Solaris Live Upgrade™ boot environments
- Solaris Volume Manager metadevices
- DHCP address assignment in a shared-IP zone
- SSL proxy server

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

- `zschd`, a system scheduling process similar to `schd`, is used to track kernel resources associated with the zone.
- `zoneadm` is the zones administration daemon.

A zone in the ready state does not have any user processes executing in it. The primary difference between a ready zone and a running zone is that at least one process is executing in a running zone. See the `init(1M)` man page for more information.
The zoneadmd Daemon

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

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

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

Unless the zoneadmd daemon is already running, it is automatically started by zoneadm. Thus, if the daemon is not running for any reason, any invocation of zoneadm to administer the zone will restart zoneadmd.

The man page for the zoneadmd daemon is zoneadmd(1M).

The zsched Zone Scheduler

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

Zone Application Environment

The zoneadm command is used to create the zone application environment.

After a non-global zone is booted for the first time, the internal configuration of the zone must be created. The internal configuration specifies a naming service to use, the default locale and time zone, the zone's root password, and other aspects of the application environment. For more information, see "Internal Zone Configuration" on page 290 and "Performing the Initial Internal Zone Configuration" on page 294. Note that the default locale and time zone for a zone can be configured independently of the global settings.
About Halting, Rebooting, and Uninstalling Zones

This section provides an overview of the procedures for halting, rebooting, uninstalling, and cloning zones.

Halting a Zone

The `zoneadm halt` command is used to remove both the application environment and the virtual platform for a zone. The zone is then brought back to the installed state. All processes are killed, devices are unconfigured, network interfaces are destroyed, file systems are unmounted, and the kernel data structures are destroyed.

The `halt` command does not run any shutdown scripts within the zone. To shut down a zone, see “How to Use zlogin to Shut Down a Zone” on page 301.

If the halt operation fails, see “Zone Does Not Halt” on page 407.

Rebooting a Zone

The `zoneadm reboot` command is used to reboot a zone. The zone is halted and then booted again. The zone ID will change when the zone is rebooted.

Zone Boot Arguments

Zones support the following boot arguments used with the `zoneadm boot` and `reboot` commands:

- `-i altinit`
- `-m smf_options`
- `-s`

The following definitions apply:

- `-i altinit` Selects an alternative executable to be the first process. `altinit` must be a valid path to an executable. The default first process is described in `init(1M)`.

- `-m smf_options` Controls the boot behavior of SMF. There are two categories of options, recovery options and messages options. Message options determine the type and number of messages that displays during boot. Service options determine the services that are used to boot the system.

  Recovery options include the following:

  `debug` Prints standard per-service output and all `svc.startd` messages to log.
milestone=milestone  Boot to the subgraph defined by the given milestone. Legitimate milestones are none, single-user, multi-user, multi-user-server, and all.

Message options include the following:

- quiet  Prints standard per-service output and error messages requiring administrative intervention
- verbose Prints standard per-service output and messages providing more information.
- s  Boots only to milestone svc:/milestone/single-user:default. This milestone is equivalent to init level s.

For usage examples, see "How to Boot a Zone" on page 278 and "How to Boot a Zone in Single-User Mode" on page 280.

For information on the Solaris service management facility (SMF) and init, see Chapter 16, "Managing Services (Overview)," in System Administration Guide: Basic Administration, svc.startd(1M) and init(1M).

**Zone autoboot**

If you set the autoboot resource property in a zone's configuration to true, that zone is automatically booted when the global zone is booted. The default setting is false.

Note that for zones to autoboot, the zones service svc:/system/zones:default must also be enabled.

**Uninstalling a Zone**

The zoneadm uninstall command is used to uninstall all of the files under the zone's root file system. Before proceeding, the command prompts you to confirm the action, unless the -F (force) option is also used. Use the uninstall command with caution, because the action is irreversible.
About Cloning Non-Global Zones

Cloning allows you to copy an existing configured and installed zone on your system to rapidly provision a new zone on the same system. Note that at a minimum, you must reset properties and resources for the components that cannot be identical for different zones. Thus, the zone path must always be changed. In addition, for a shared-IP zone, the IP addresses in any net resources must be different. For an exclusive-IP zone, the physical property of any net resources must be different.

- Cloning a zone is a faster way to install a zone.
- The new zone will include any changes that have been made to customize the source zone, such as added packages or file modifications.

When the source zone path and the target zone path both reside on ZFS and are in the same pool, the zoneadm clone command automatically uses ZFS to clone the zone. When using ZFS clone, the data is not actually copied until it is modified. Thus, the initial clone takes very little time. The zoneadm command takes a ZFS snapshot of the source zone path, and sets up the target zone path. The system names the snapshot SUNWzoneX, where X is a unique ID used to distinguish between multiple snapshots. The zone path of the destination zone is used to name the ZFS clone. A software inventory is performed so that a snapshot used at a future time can be validated by the system. To clone a source zone multiple times, the zoneadm command allows you to specify that an existing snapshot should be used. The system validates that the existing snapshot is usable on the target.

You cannot use manual snapshots, such as the type described in “Creating and Destroying ZFS Snapshots” in Solaris ZFS Administration Guide. This type of snapshot lacks the data to perform a validation.

You might want to clone a source zone many times but not want to have a new snapshot for each clone. The -s parameter to the clone subcommand allows you to specify that an existing snapshot taken from a previous clone should be used. See “How to Clone a Zone from an Existing Snapshot” on page 285.

Because a snapshot’s contents represent a zone from a point in the past, it is possible that the system has been updated in some way, such as by patching or upgrading, since the snapshot was taken. The fact that the zone was upgraded could render the snapshot invalid for use as a zone on the present-day system.

Note – You can specify that a ZFS zone path be copied instead of ZFS cloned, even though the source could be cloned in this way.

See “Cloning a Non-Global Zone on the Same System” on page 284 for more information.
Installing, Booting, Halting, Uninstalling, and Cloning Non-Global Zones (Tasks)

This chapter describes how to install and boot a non-global zone. A method for using cloning to install a zone on the same system is also provided. Other tasks associated with installation, such as halting, rebooting, and uninstalling zones, are addressed. The procedure to completely delete a zone from a system is also included.

For general information about zone installation and related operations, see Chapter 18, “About Installing, Halting, Uninstalling, and Cloning Non-Global Zones (Overview).”

For information about \textvisiblespace\textvisiblespace lx branded zone installation and cloning, see Chapter 32, “About Installing, Booting, Halting, Cloning, and Uninstalling \textvisiblespace lx\textvisiblespace Branded Zones (Overview),” and Chapter 33, “Installing, Booting, Halting, Uninstalling and Cloning \textvisiblespace lx\textvisiblespace Branded Zones (Tasks).”

Zone Installation (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional) Verify a configured zone prior to installing the zone.</td>
<td>Ensure that a zone meets the requirements for installation. If you skip this procedure, the verification is performed automatically when you install the zone.</td>
<td>“(Optional) How to Verify a Configured Zone Before It Is Installed” on page 274</td>
</tr>
<tr>
<td>Install a configured zone.</td>
<td>Install a zone that is in the configured state.</td>
<td>“How to Install a Configured Zone” on page 275</td>
</tr>
<tr>
<td>Obtain the universally unique identifier (UUID) for the zone.</td>
<td>This separate identifier, assigned when the zone is installed, is an alternate way to identify a zone.</td>
<td>“How to Obtain the UUID of an Installed Non-Global Zone” on page 276</td>
</tr>
<tr>
<td>(Optional) Transition an installed zone to the ready state.</td>
<td>You can skip this procedure if you want to boot the zone and use it immediately.</td>
<td>“(Optional) How to Transition the Installed Zone to the Ready State” on page 278</td>
</tr>
</tbody>
</table>
Installing and Booting Zones

Use the zoneadm command described in the zoneadm(1M) man page to perform installation tasks for a non-global zone. You must be the global administrator to perform the zone installation. The examples in this chapter use the zone name and zone path established in “Configuring, Verifying, and Committing a Zone” on page 249.

▼ (Optional) How to Verify a Configured Zone Before It Is Installed

You can verify a zone prior to installing it. One of the checks performed is a check for sufficient disk size. If you skip this procedure, the verification is performed automatically when you install the zone.

You must be the global administrator in the global zone to perform this procedure.

1. Become superuser, or assume the Primary Administrator role.

   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. Verify a configured zone named my-zone by using the -z option with the name of the zone and the verify subcommand.

   global# zoneadm -z my-zone verify

   This message regarding verification of the zone path will be displayed:

   Warning: /export/home/my-zone does not exist, so it cannot be verified.
   When 'zoneadm install' is run, 'install' will try to create
   /export/home1/my-zone, and 'verify' will be tried again,
   but the 'verify' may fail if:
the parent directory of /export/home/my-zone is group- or other-writable or
/export/home1/my-zone overlaps with any other installed zones.

However, if an error message is displayed and the zone fails to verify, make the corrections specified in the message and try the command again.

If no error messages are displayed, you can install the zone.

▼ How to Install a Configured Zone

This procedure is used to install a configured non-global zone.

You must be the global administrator in the global zone to perform this procedure.

Note – In Step 2, if the zone path is on ZFS, the zoneadm install command automatically creates a ZFS file system (dataset) for the zone path when the zone is installed. You can block this action by including the -x nodataset parameter.

1 Become superuser, or assume the Primary Administrator role.

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Install the configured zone my-zone by using the zoneadm command with the install option.

- Install the zone, automatically creating a ZFS file system if the zone path is on ZFS.
  global# zoneadm -z my-zone install
  The system will display:
  
  A ZFS file system has been created for this zone.
  
- Install the zone that has a zone path on ZFS, but do not automatically create the ZFS file system.
  global# zoneadm -z my-zone install -x nodataset
  You will see various messages as the files and directories needed for the zone’s root file system are installed under the zone’s root path.

3 (Optional) If an error message is displayed and the zone fails to install, type the following to get the zone state:

  global# zoneadm -z my-zone list -v

  ▪ If the state is listed as configured, make the corrections specified in the message and try the zoneadm install command again.
If the state is listed as incomplete, first execute this command:

```
global# zoneadm -z my-zone uninstall
```

Then make the corrections specified in the message, and try the `zoneadm install` command again.

When the installation completes, use the `list` subcommand with the `-i` and `-v` options to list the installed zones and verify the status.

```
global# zoneadm list -iv
```

You will see a display that is similar to the following:

```
ID  NAME   STATUS   PATH                        BRAND  IP
0   global running /                          native shared
    my-zone installed /export/home/my-zone native shared
```

Troubleshooting If a zone installation is interrupted or fails, the zone is left in the incomplete state. Use `uninstall -F` to reset the zone to the configured state.

Next Steps This zone was installed with the minimal network configuration described in Chapter 17, “Managing Services (Tasks),” in System Administration Guide: Basic Administration by default. You can switch to the open network configuration, or enable or disable individual services, when you log in to the zone. See "Switching the Non-Global Zone to a Different Networking Service Configuration" on page 301 for details.

How to Obtain the UUID of an Installed Non-Global Zone

A universally unique identifier (UUID) is assigned to a zone when it is installed. The UUID can be obtained by using `zoneadm` with the `list` subcommand and the `-p` option. The UUID is the fifth field of the display.

- View the UUIDs for zones that have been installed.

```
global# zoneadm list -p
```

You will see a display similar to the following:

```
0:global:running::native
6:my-zone:running:/export/home/my-zone:61901255-35cf-40d6-d501-f37dc84eb504:native
```

Example 19–1 How to Use the Zone UUID in a Command

```
global# zoneadm -z my-zone -u 61901255-35cf-40d6-d501-f37dc84eb504 list -v
```
If both -u uuid-match and -z zonename are present, the match is done based on the UUID first. If a zone with the specified UUID is found, that zone is used, and the -z parameter is ignored. If no zone with the specified UUID is found, then the system searches by the zone name.

More Information  About the UUID

Zones can be uninstalled and reinstalled under the same name with different contents. Zones can also be renamed without the contents being changed. For these reasons, the UUID is a more reliable handle than the zone name.

See Also  For more information, see zoneadm(1M) and libuuid(3LIB).

▼  How to Mark an Installed Non-Global Zone Incomplete

If administrative changes on the system have rendered a zone unusable or inconsistent, it is possible to change the state of an installed zone to incomplete.

You must be the global administrator in the global zone to perform this procedure.

1  Become superuser, or assume the Primary Administrator role.
To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2  Mark the zone testzone incomplete.

   global# zoneadm -z testzone mark incomplete

3  Use the list subcommand with the -i and -v options to verify the status.

   global# zoneadm list -iv

You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>my-zone</td>
<td>installed</td>
<td>/export/home/my-zone</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>testzone</td>
<td>incomplete</td>
<td>/export/home/testzone</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>

More Information  Marking a Zone Incomplete

The -R root option can be used with the mark and list subcommands of zoneadm to specify an alternate boot environment. See zoneadm(1M) for more information.
Note – Marking a zone incomplete is irreversible. The only action that can be taken on a zone marked incomplete is to uninstall the zone and return it to the configured state. See “How to Uninstall a Zone” on page 283.

▼ (Optional) How to Transition the Installed Zone to the Ready State

Transitioning into the ready state prepares the virtual platform to begin running user processes. Zones in the ready state do not have any user processes executing in them.

You can skip this procedure if you want to boot the zone and use it immediately. The transition through the ready state is performed automatically when you boot the zone.

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Use the zoneadm command with the -z option, the name of the zone, which is my-zone, and the ready subcommand to transition the zone to the ready state.

   global# zoneadm -z my-zone ready

3 At the prompt, use the zoneadm list command with the -v option to verify the status.

   global# zoneadm list -v

   You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>1</td>
<td>my-zone</td>
<td>ready</td>
<td>/export/home/my-zone</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>

   Note that the unique zone ID 1 has been assigned by the system.

▼ How to Boot a Zone

Booting a zone places the zone in the running state. A zone can be booted from the ready state or from the installed state. A zone in the installed state that is booted transparently transitions through the ready state to the running state. Zone login is allowed for zones in the running state.
Tip – Note that you perform the internal zone configuration when you initially log in to the zone. This is described in “Performing the Initial Internal Zone Configuration” on page 294.

If you plan to use an /etc/sysidcfg file to perform initial zone configuration, as described in “How to Use an /etc/sysidcfg File to Perform the Initial Zone Configuration” on page 296, create the sysidcfg file and place it the zone’s /etc directory before you boot the zone.

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Use the zoneadm command with the -z option, the name of the zone, which is my-zone, and the boot subcommand to boot the zone.

   global# zoneadm -z my-zone boot

3 When the boot completes, use the list subcommand with the -v option to verify the status.

   global# zoneadm list -v

   You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>1</td>
<td>my-zone</td>
<td>running</td>
<td>/export/home/my-zone</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>

Example 19–2 Specifying Boot Arguments for Zones

Boot a zone using the -m verbose option:

   global# zoneadm -z my-zone boot -- -m verbose

Reboot a zone using the -m verbose boot option:

   global# zoneadm -z my-zone reboot -- -m verbose

Zone administrator reboot of the zone my-zone, using the -m verbose option:

   my-zone# reboot -- -m verbose

Troubleshooting If a message indicating that the system was unable to find the netmask to be used for the IP address specified in the zone’s configuration displays, see “netmasks Warning Displayed When Booting Zone” on page 406. Note that the message is only a warning and the command has succeeded.
How to Boot a Zone in Single-User Mode

You must be the global administrator in the global zone to perform this procedure.

1 **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2 **Boot the zone in single-user mode.**
   
   ```
   global# zoneadm -z my-zone boot -- -s
   ```

Where to Go From Here

To log in to the zone and perform the initial internal configuration, see Chapter 20, “Non-Global Zone Login (Overview),” and Chapter 21, “Logging In to Non-Global Zones (Tasks).”

### Halting, Rebooting, Uninstalling, Cloning, and Deleting Non-Global Zones (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halt a zone.</td>
<td>The halt procedure is used to remove both the application environment and the virtual platform for a zone. The procedure returns a zone in the ready state to the installed state. To cleanly shut down a zone, see &quot;How to Use zlogin to Shut Down a Zone&quot; on page 301.</td>
<td>“How to Halt a Zone” on page 281</td>
</tr>
<tr>
<td>Reboot a zone.</td>
<td>The reboot procedure halts the zone and then boots it again.</td>
<td>“How to Reboot a Zone” on page 282</td>
</tr>
<tr>
<td>Uninstall a zone.</td>
<td>This procedure removes all of the files in the zone's root file system. Use this procedure with caution. The action is irreversible.</td>
<td>“How to Uninstall a Zone” on page 283</td>
</tr>
</tbody>
</table>
Halting, Rebooting, and Uninstalling Zones

How to Halt a Zone

The halt procedure is used to remove both the application environment and the virtual platform for a zone. To cleanly shut down a zone, see “How to Use zlogin to Shut Down a Zone” on page 301.

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 List the zones running on the system.

```
global# zoneadm list -v
```

You will see a display that is similar to the following:

```
ID  NAME   STATUS  PATH         BRAND  IP
 0  global  running /  native  shared
 1  my-zone running /export/home/my-zone  native  shared
```

3 Use the `zoneadm` command with the `-z` option, the name of the zone, for example, `my-zone`, and the `halt` subcommand to halt the given zone.

```
global# zoneadm -z my-zone halt
```

4 List the zones on the system again, to verify that `my-zone` has been halted.

```
global# zoneadm list -iv
```

You will see a display that is similar to the following:

```
ID  NAME   STATUS  PATH         BRAND  IP
 0  global  running /  native  shared
```
- my-zone installed /export/home/my-zone native shared

5 Boot the zone if you want to restart it.

```
global# zoneadm -z my-zone boot
```

Troubleshooting If the zone does not halt properly, see “Zone Does Not Halt” on page 407 for troubleshooting tips.

▶ How to Reboot a Zone

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.

To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 List the zones running on the system.

```
global# zoneadm list -v
```

You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>1</td>
<td>my-zone</td>
<td>running</td>
<td>/export/home/my-zone</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>

3 Use the `zoneadm` command with the `-z reboot` option to reboot the zone my-zone.

```
global# zoneadm -z my-zone reboot
```

4 List the zones on the system again to verify that my-zone has been rebooted.

```
global# zoneadm list -v
```

You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>2</td>
<td>my-zone</td>
<td>running</td>
<td>/export/home/my-zone</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>

Tip – Note that the zone ID for my-zone has changed. The zone ID generally changes after a reboot.
How to Uninstall a Zone

Caution – Use this procedure with caution. The action of removing all of the files in the zone’s root file system is irreversible.

The zone cannot be in the running state. The uninstall operation is invalid for running zones.

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 List the zones on the system.

global# zoneadm list -v

You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>my-zone</td>
<td>installed</td>
<td>/export/home/my-zone</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>

3 Use the zoneadm command with the -z uninstall option to remove the zone my-zone.
You can also use the -F option to force the action. If this option is not specified, the system will prompt for confirmation.

global# zoneadm -z my-zone uninstall -F

Note that when you uninstall a zone that has its own ZFS file system for the zone path, the ZFS file system is destroyed.

4 List the zones on the system again, to verify that my-zone is no longer listed.

global# zoneadm list -iv

You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>

Troubleshooting If a zone uninstall is interrupted, the zone is left in the incomplete state. Use the zoneadm uninstall command to reset the zone to the configured state.

Use the uninstall command with caution because the action is irreversible.
Cloning a Non-Global Zone on the Same System

Cloning is used to provision a new zone on a system by copying the data from a source zonepath to a target zonepath.

When the source zonepath and the target zonepath both reside on ZFS and are in the same pool, the zoneadm clone command automatically uses ZFS to clone the zone. However, you can specify that the ZFS zonepath be copied and not ZFS cloned.

How to Clone a Zone

You must configure the new zone before you can install it. The parameter passed to the zoneadm create subcommand is the name of the zone to clone. This source zone must be halted.

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Halt the source zone to be cloned, which is my-zone in this procedure.
   global# zoneadm -z my-zone halt

3 Start configuring the new zone by exporting the configuration of the source zone my-zone to a file, for example, master.
   global# zonecfg -z my-zone export -f /export/zones/master

   Note – You can also create the new zone configuration using the procedure “How to Configure the Zone” on page 249 instead of modifying an existing configuration. If you use this method, skip ahead to Step 6 after you create the zone.

4 Edit the file master. Set different properties and resources for the components that cannot be identical for different zones. For example, you must set a new zonepath. For a shared-IP zone, the IP addresses in any net resources must be changed. For an exclusive-IP zone, the physical property of any net resources must be changed.

5 Create the new zone, zone1, by using the commands in the file master.
   global# zonecfg -z zone1 -f /export/zones/master

6 Install the new zone, zone1, by cloning my-zone.
   global# zoneadm -z zone1 clone my-zone
The system displays:

Cloning zonepath /export/home/my-zone...

If the source zonepath is on a ZFS pool, for example, zeepool, the system displays:

Cloning snapshot zeepool/zones/my-zone@SUNWzone1

Instead of copying, a ZFS clone has been created for this zone.

### 7 List the zones on the system.

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td></td>
<td>my-zone</td>
<td>installed</td>
<td>/export/home/my-zone</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td></td>
<td>zone1</td>
<td>installed</td>
<td>/export/home/zone1</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>

#### More Information

**When a Source zonepath on a ZFS File System Is Cloned**

When the `zoneadm` command clones a source zonepath that is on its own ZFS file system, the following actions are performed:

- The `zoneadm` command takes a software inventory.
- The `zoneadm` command takes a ZFS snapshot and names it `SUNWzoneX`, for example, `SUNWzone1`.
- The `zoneadm` command uses ZFS clone to clone the snapshot.

### How to Clone a Zone from an Existing Snapshot

You can clone a source zone multiple times from an existing snapshot that was originally taken when you cloned a zone.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in *System Administration Guide: Basic Administration*.

2. **Configure the zone** zone2.

3. **Specify that an existing snapshot be used to create** new-zone2.

   ```
   global# zoneadm -z zone2 clone -s zeepool/zones/my-zone@SUNWzone1 my-zone
   ```

   The system displays:

   ```
   Cloning snapshot zeepool/zones/my-zone@SUNWzone1
   ```
The `zoneadm` command validates the software from the snapshot `SUNWzone1`, and clones the snapshot.

### List the zones on the system.

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>my-zone</td>
<td>installed</td>
<td>/zeepool/zones/my-zone</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>zone1</td>
<td>installed</td>
<td>/zeepool/zones/zone1</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>zone2</td>
<td>installed</td>
<td>/zeepool/zones/zone2</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>

#### How to Use Copy Instead of ZFS Clone

Use this procedure to prevent the automatic cloning of a zone on a ZFS file system by specifying that the zone path should be copied instead.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Specify that the zone path on ZFS be copied and not ZFS cloned.**
   
   ```
   global# zoneadm -z zone1 clone -m copy my-zone
   ```

### Deleting a Non-Global Zone From the System

The procedure described in this section completely deletes a zone from a system.

#### How to Remove a Non-Global Zone

1. **Shut down the zone** `my-zone`.
   
   ```
   global# zlogin my-zone shutdown -y -g0 -i0
   ```

2. **Remove the root file system for** `my-zone`.
   
   ```
   global# zoneadm -z my-zone uninstall -F
   ```

3. **Delete the configuration for** `my-zone`.
   
   ```
   global# zonecfg -z my-zone delete -F
   ```
List the zones on the system, to verify that my-zone is no longer listed.

global# zoneadm list -iv

You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>
This chapter discusses logging into zones from the global zone.

The following topics are covered in this chapter:

- “zlogin Command” on page 289
- “Internal Zone Configuration” on page 290
- “Non-Global Zone Login Methods” on page 290
- “Interactive and Non-Interactive Modes” on page 292
- “Failsafe Mode” on page 291
- “Remote Login” on page 291

For procedures and usage information, see Chapter 21, "Logging In to Non-Global Zones (Tasks)."

For information about logging into \x branded zones, see Chapter 34, "Logging In to \x Branded Zones (Tasks)."

**zlogin Command**

After you install a zone, you must log in to the zone to complete its application environment. You might log in to the zone to perform administrative tasks as well. Unless the -C option is used to connect to the zone console, logging in to a zone using zlogin starts a new task. A task cannot span two zones.

The `zlogin` command is used to log in from the global zone to any zone that is in the running state or the ready state.

**Note** – Only the `zlogin` command with the -C option can be used to log in to a zone that is not in the running state.
As described in “How to Use Non-Interactive Mode to Access a Zone” on page 299, you can use the `zlogin` command in non-interactive mode by supplying a command to run inside a zone. However, the command or any files the command acts upon cannot reside on NFS. The command will fail if any of its open files or any portion of its address space resides on NFS. The address space includes the command executable itself and the command’s linked libraries.

The `zlogin` command can only be used by the global administrator operating in the global zone. See the `zlogin(1)` man page for more information.

**Internal Zone Configuration**

After installation, the zone is in an unconfigured state. The zone does not have an internal configuration for naming services, its locale and time zone have not been set, and various other configuration tasks have not been performed. Therefore, the `sysidtool` programs are run the first time a zone is booted. For more information, see the `sysidtool(1M)` man page.

Two methods are available for performing the required configuration:

- Zone console login, which initiates a series of questions from the system. Be prepared to respond to the following:
  - Language
  - Type of terminal being used
  - Host name
  - Security policy (Kerberos or standard UNIX)
  - Naming service type (`None` is a valid response)
  - Naming service domain
  - Name server
  - Default time zone
  - Root password

  The procedure is described in “Performing the Initial Internal Zone Configuration” on page 294.

- An `/etc/sysidcfg` file, which you can create and place inside the zone before you boot the zone for the first time. See the `sysidcfg(4)` man page for more information.

**Non-Global Zone Login Methods**

This section describes the methods you can use to log in to a zone.

**Zone Console Login**

Each zone maintains a virtual console, `/dev/console`. Performing actions on the console is referred to as console mode. Console login to a zone is available when the zone is in the installed
state. The zone console is closely analogous to a serial console on a system. Connections to the console persist across zone reboots. To understand how console mode differs from a login session such as telnet, see “Remote Login” on page 291.

The zone console is accessed by using the zlogin command with the -C option and the zonename. The zone does not have to be in the running state.

Processes inside the zone can open and write messages to the console. If the zlogin -C process exits, another process can then access the console.

**User Login Methods**

To log in to the zone with a username, use the zlogin command with the -l option, the user name, and the zonename. For example, the administrator of the global zone can log in as a normal user in the non-global zone by specifying the -l option to zlogin:

```
global# zlogin -l user zonename
```

To log in as user root, use the zlogin command without options.

**Failsafe Mode**

If a login problem occurs and you cannot use the zlogin command or the zlogin command with the -C option to access the zone, an alternative is provided. You can enter the zone by using the zlogin command with the -S (safe) option. Only use this mode to recover a damaged zone when other forms of login are not succeeding. In this minimal environment, it might be possible to diagnose why the zone login is failing.

**Remote Login**

The ability to remotely log in to a zone is dependent on the selection of network services that you establish. By default, a non-global zone is installed with the limited networking configuration (/var/svc/profile/generic_limited_net.xml), and only the ssh login is enabled. Logins through rlogin and telnet can be added if needed, either by using the netservices command to switch the zone to the open networking configuration or by enabling the services using SMF.

For more information about changing the network profile or using SMF commands to add services to zones, see “Switching the Non-Global Zone to a Different Networking Service Configuration” on page 301. For more information about login commands, see rlogin(1), ssh(1), and telnet(1)
Interactive and Non-Interactive Modes

Two other methods for accessing the zone and for executing commands inside the zone are also provided by the `zlogin` command. These methods are interactive mode and non-interactive mode.

**Interactive Mode**

In interactive mode, a new pseudo-terminal is allocated for use inside the zone. Unlike console mode, in which exclusive access to the console device is granted, an arbitrary number of `zlogin` sessions can be open at any time in interactive mode. Interactive mode is activated when you do not include a command to be issued. Programs that require a terminal device, such as an editor, operate correctly in this mode.

**Non-Interactive Mode**

Non-interactive mode is used to run shell-scripts which administer the zone. Non-interactive mode does not allocate a new pseudo-terminal. Non-interactive mode is enabled when you supply a command to be run inside the zone.
This chapter provides procedures for completing the configuration of an installed zone, logging into a zone from the global zone, and shutting down a zone. This chapter also shows how to use the zonename command to print the name of the current zone.

For an introduction to the zone login process, see Chapter 20, “Non-Global Zone Login (Overview).”

For information about logging into \lx branded zones, see Chapter 34, “Logging In to \lx Branded Zones (Tasks).”

### Initial Zone Boot and Zone Login Procedures (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform the internal configuration.</td>
<td>Log in to the zone console or use an /etc/sysidcfg file to perform the initial zone configuration.</td>
<td>&quot;Performing the Initial Internal Zone Configuration&quot; on page 294</td>
</tr>
<tr>
<td>Log in to the zone.</td>
<td>You can log into a zone through the console, by using interactive mode to allocate a pseudo-terminal, or by supplying a command to be run in the zone. Supplying a command to be run does not allocate a pseudo-terminal. You can also log in by using failsafe mode when a connection to the zone is denied.</td>
<td>&quot;Logging In to a Zone&quot; on page 298</td>
</tr>
<tr>
<td>Exit a non-global zone.</td>
<td>Disconnect from a non-global zone.</td>
<td>&quot;How to Exit a Non-Global Zone&quot; on page 300</td>
</tr>
</tbody>
</table>
Performing the Initial Internal Zone Configuration

You must configure the zone using one of the following methods:

- Log into the zone and configure it as described in “Internal Zone Configuration” on page 290.
- Configure the zone using an /etc/sysidcfg file as described in “How to Use an /etc/sysidcfg File to Perform the Initial Zone Configuration” on page 296.

**Tip** – After you have performed the internal configuration, it is a good idea to make a copy of the non-global zone’s configuration. You can use this backup to restore the zone in the future. As superuser or Primary Administrator, print the configuration for the zone my-zone to a file. This example uses a file named my-zone.config.

```
global# zonecfg -z my-zone export > my-zone.config
```

See “How to Restore an Individual Non-Global Zone” on page 402 for more information.

**How to Log In to the Zone Console to Perform the Internal Zone Configuration**

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. **Use the zlogin command with the -C option and the name of the zone, my-zone in this procedure.**
   
   global# zlogin -C my-zone

3. **From another terminal window, boot the zone.**
   
   global# zoneadm -z my-zone boot
You will see a display similar to the following in the zlogin window:

[NOTICE: Zone booting up]

4 The first time you log in to the console, you are prompted to answer a series of questions. Your screen will look similar to this:

SunOS Release 5.10 Version Generic 64-bit
Copyright 1983-2006 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Hostname: my-zone
Loading smf(5) service descriptions: 114/114
Select a Language
1. English
2. es
3. fr

Please make a choice (1 - 3), or press h or ? for help:

Select a Locale
1. English (C - 7-bit ASCII)
2. Canada (English) (UTF-8)
5. U.S.A. (en.US.ISO8859-15)
6. Go Back to Previous Screen

Please make a choice (1 - 7), or press h or ? for help:

What type of terminal are you using?
1) ANSI Standard CRT
2) DEC VT52
3) DEC VT100
4) Heathkit 19
5) Lear Siegler ADM31
6) PC Console
7) Sun Command Tool
8) Sun Workstation
9) Televideo 910
10) Televideo 925
11) Wyse Model 50
12) X Terminal Emulator (xterms)
13) CDE Terminal Emulator (dtterm)
14) Other

Type the number of your choice and press Return:
13
For the complete list of questions you must answer, see “Internal Zone Configuration” on page 290.

(5 Optional) If you are not using two windows as described in step 3, you might have missed the initial prompt for configuration information. If you see the following system message at zone login instead of a prompt:

[connected to zone zonename console]

Press Return to display the prompt again.

If you enter an incorrect response and try to restart the configuration, you might experience difficulty when you attempt the process again. This occurs because the sysidtools can store your previous responses.

If this happens, use the following workaround from the global zone to restart the configuration process.

global# zlogin -S zonename /usr/sbin/sys-unconfig

For more information on the sys-unconfig command, see the sys-unconfig(1M) man page.

▼ How to Use an /etc/sysidcfg File to Perform the Initial Zone Configuration

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 From the global zone, change directories to the non-global zone's /etc directory:

global# cd /export/home/my-zone/root/etc

3 Create the sysidcfg file and place it in this directory.

The file will look similar to the following:

- For a shared-IP zone:

```plaintext
system_locale=C
terminal=dtterm
network_interface=primary {
    hostname=my-zone
}
```
security_policy=NONE
name_service=NIS {
    domain_name=special.example.com
    name_server=bird(192.168.112.3)
}
nfs4_domain=domain
timezone=US/Central
root_password=m4qtoWN

- For an exclusive-IP zone with a static IP configuration:
  system_locale=C
terminal=dtterm
network_interface=primary {
    hostname=my-zone
default_route=10.10.10.1
ip_address=10.10.10.13
netmask=255.255.255.0
}
nfs4_domain=domain
timezone=US/Central
root_password=m4qtoWN

- For an exclusive-IP zone with DHCP and IPv6 option:
  system_locale=C
terminal=dtterm
network_interface=primary {
    dhcp protocol_ipv6=yes
}
security_policy=NONE
name_service=DNS {
    domain_name=example.net
    name_server=192.168.224.11,192.168.224.33
}
nfs4_domain=domain
timezone=US/Central
root_password=m4qtoWN

4 If you are running an earlier release and do not have the nfs4_domain parameter in your file, by default, a separate module will request the NFSv4 domain parameter used by the nfsmapid command. To complete a hands-off initial zone configuration, edit the file default/nfs, uncomment the NFSMAPID_DOMAIN parameter, and set the domain to the desired NFSv4 domain:
   global# vi default/nfs
   ...
   ...
   NFSMAPID_DOMAIN=domain
Create the file `.NFS4inst_state.domain` in this directory to indicate that the NFSv4 domain has been set:

```
global# touch .NFS4inst_state.domain
```

For more information on the NFSv4 domain parameter, see the `nfsmapid(1M)` man page.

5) **Boot the zone.**

**See Also** See the `sysidcfg(4)` man page for more information.

---

# Logging In to a Zone

Use the `zlogin` command to log in from the global zone to any zone that is running or in the ready state. See the `zlogin(1)` man page for more information.

You can log in to a zone in various ways, as described in the following procedures. You can also log in remotely, as described in “Remote Login” on page 291.

---

## How to Log In to the Zone Console

You must be the global administrator in the global zone to perform this procedure.

1) **Become superuser, or assume the Primary Administrator role.**

   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration.*

2) **Use the `zlogin` command with the `-C` option and the name of the zone, for example, `my-zone`.

   ```
global# zlogin -C my-zone
   ```

**Note** – If you start the `zlogin` session immediately after issuing the `zoneadm boot` command, boot messages from the zone will display:

```
SunOS Release 5.10 Version Generic 64-bit
Copyright 1983-2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
starting rpc services: rpcbind done.
syslog service starting.
The system is ready.
```
When the zone console displays, log in as root, press Return, and type the root password when prompted.

```
my-zone console login: root
Password:
```

### How to Use Interactive Mode to Access a Zone

In interactive mode, a new pseudo-terminal is allocated for use inside the zone.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in *System Administration Guide: Basic Administration*.

2. **From the global zone, log in to the zone, for example, my-zone.**
   
   ```
   global# zlogin my-zone
   ```
   
   Information similar to the following will display:

   ```
   [Connected to zone 'my-zone' pts/2]
   Last login: Wed Jul  3 16:25:00 on console
   Sun Microsystems Inc. SunOS 5.10 Generic July 2004
   ```

3. **Type exit to close the connection.**
   
   You will see a message similar to the following:

   ```
   [Connection to zone 'my-zone' pts/2 closed]
   ```

### How to Use Non-Interactive Mode to Access a Zone

Non-interactive mode is enabled when the user supplies a command to be run inside the zone. Non-interactive mode does not allocate a new pseudo-terminal.

Note that the command or any files that the command acts upon cannot reside on NFS.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in *System Administration Guide: Basic Administration*. 

---

Chapter 21 • Logging In to Non-Global Zones (Tasks) 299
2 From the global zone, log in to the my-zone zone and supply a command name. The command zonename is used here.

```
global# zlogin my-zone zonename
```

You will see the following output:

```
my-zone
```

- **How to Exit a Non-Global Zone**

  To disconnect from a non-global zone, use one of the following methods.

  - To exit the zone non-virtual console:
    ```
    zonename# exit
    ```

  - To disconnect from a zone virtual console, use the tilde (~) character and a period:
    ```
    zonename# ~.
    ```

    Your screen will look similar to this:

    ```
    [Connection to zone 'lx-zone' pts/6 closed]
    ```

  See Also For more information about zlogin command options, see the `zlogin(1)` man page.

- **How to Use Failsafe Mode to Enter a Zone**

  When a connection to the zone is denied, the `zlogin` command can be used with the `-S` option to enter a minimal environment in the zone.

  You must be the global administrator in the global zone to perform this procedure.

1 **Become superuser, or assume the Primary Administrator role.**

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 **From the global zone, use the `zlogin` command with the `-S` option to access the zone, for example, `my-zone`.**

```
global# zlogin -S my-zone
```
How to Use `zlogin` to Shut Down a Zone

**Note** – Running `init 0` in the global zone to cleanly shut down a Solaris system also runs `init 0` in each of the non-global zones on the system. Note that `init 0` does not warn local and remote users to log off before the system is taken down.

Use this procedure to cleanly shut down a zone. To halt a zone without running shutdown scripts, see "How to Halt a Zone" on page 281.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Log in to the zone to be shut down, for example, `my-zone`, and specify `shutdown` as the name of the utility and `init 0` as the state.**

   ```
   global# zlogin my-zone shutdown -y -g0 -i0
   ```

   Your site might have its own shutdown script, tailored for your specific environment.

   **Note** – You cannot use the `shutdown` command to place the zone in single-user state at this time. See 6214427 for more information.

**More Information**

Using `shutdown` in Non-Interactive Mode

You cannot use the `shutdown` command in non-interactive mode to place the zone in single-user state at this time. See 6214427 for more information.

You can use an interactive login as described in “How to Use Interactive Mode to Access a Zone” on page 299.

Switching the Non-Global Zone to a Different Networking Service Configuration

This zone was installed with the minimal networking service configuration described in Chapter 17, "Managing Services (Tasks),” in *System Administration Guide: Basic Administration*. You can switch the zone to the open networking service configuration, or enable or disable individual services in the zone.
How to Switch the Zone to the Open Networking Service Configuration

1. From the global zone, log in to the zone, for example, `my-zone`.
   ```
global# zlogin my-zone
   ```

2. Run the `netservices` command to switch the zone to the traditional open networking configuration.
   ```
my-zone# /usr/sbin/netservices open
   ```

   You will see a display similar to the following. Respond `y` to the prompt to restart `dtlogin`.

   ```
   restarting syslogd
   restarting sendmail
   dtlogin needs to be restarted. Restart now? [Y] y
   restarting dtlogin
   ```

How to Enable a Specific Service in a Zone

1. From the global zone, log in to the zone, for example, `my-zone`.
   ```
global# zlogin my-zone
   ```

2. Run the `svcadm` command to enable `rlogin`.
   ```
my-zone# svcadm enable svc:/network/login:rlogin
   ```

3. List the services to verify that `rlogin` is enabled.
   ```
my-zone# svcs -a
   ```

   ```
online 14:01:08 svc:/network/login:rlogin
   ```
Printing the Name of the Current Zone

The `zonename` command described in the `zonename(1)` man page prints the name of the current zone. The following example shows the output when `zonename` is used in the global zone.

```
# zonename
global
```
This chapter describes how to:

- Move an existing non-global zone to a new location on the same machine.
- Validate what will happen in a non-global zone migration before the actual migration is performed.
- Migrate an existing non-global zone to a new machine.

If the new host has later versions of the zone-dependent packages or the associated patches, using `zoneadm attach` with the `-u` option updates those packages within the zone to match the new host. If the new host has a mixture of higher and lower version patches as compared to the source host, then an update during the attach operation is not allowed.

The `-u` option enables migration between machine classes, such as from `sun4u` to `sun4v`.

The `-b` option can be used to specify patches, either official or Interim Diagnostics/Relief (IDR), to be backed out of the zone before the update. Multiple `-b` options can be specified. If any of the patches cannot be backed out for any reason, then the `attach` will fail and none of the patches will be backed out.

This option only applies to zone brands using SVR4 packaging.

For information on moving and migrating \lx branded zones, see Chapter 35, “Moving and Migrating \lx Branded Zones (Tasks).”

**Moving a Non-Global Zone**

This procedure is used to move the zone to a new location on the same system by changing the `zonepath`. The zone must be halted. The new `zonepath` must be on a local file system. The normal `zonepath` criteria described in “Resource and Property Types” on page 229 apply.
**How to Move a Zone**

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Halt the zone to be moved, db-zone in this procedure.**
   ```
   global# zoneadm -z db-zone halt
   ```

3. **Use the zoneadm command with the move subcommand to move the zone to a new zonepath, /export/zones/db-zone.**
   ```
   global# zoneadm -z db-zone move /export/zones/db-zone
   ```

4. **Verify the path.**
<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>my-zone</td>
<td>installed</td>
<td>/export/home/my-zone</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>db-zone</td>
<td>installed</td>
<td>/export/zones/db-zone</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>

**Migrating a Non-Global Zone to a Different Machine**

Note that you can do a trial run of a zone migration before you actually move the zone to a different machine. For more information, see “About Validating a Zone Migration Before the Migration Is Performed” on page 311.

**About Migrating a Zone**

The `zonecfg` and `zoneadm` commands can be used to migrate an existing non-global zone from one system to another. The zone is halted and detached from its current host. The zonepath is moved to the target host, where it is attached.

The following requirements apply to zone migration:

- The global zone on the target system must be running the same Solaris release as the original host.
- To ensure that the zone will run properly, the target system must have the same or later versions of the following required operating system packages and patches as those installed on the original host.
  - Packages that deliver files under an `inherit-pkg-dir` resource
  - Packages where `SUNW_PKG_ALLZONES=true`
Other packages and patches, such as those for third-party products, can be different.

- If the new host has later versions of the zone-dependent packages or their associated patches, using `zoneadm attach` with the `-u` option updates those packages within the zone to match the new host. The update on attach software looks at the zone that is being migrated and determines which packages must be updated to match the new host. Only those packages are updated. The rest of the packages, and their associated patches, can vary from zone to zone. This option also enables automatic migration between sun4u and sun4v machine classes. The `-b` option can be used to specify patches to be backed out of the zone before the update.

- The host and target systems must have the same machine architecture unless the `-u` option, which can be used to migrate between sun4u and sun4v machine classes, is used.

- The `-b` option can be used to specify patches, either official or IDR, to be backed out of the zone before the update. Multiple `-b` options can be specified. If any of the patches cannot be backed out for any reason, then the `attach` will fail and none of the patches will be backed out.

This new option is brand-specific and only applies to zone brands using SVR4 packaging.

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

```
# uname -m
```

The `zoneadm detach` process creates the information necessary to attach the zone on a different system. The `zoneadm attach` process verifies that the target machine has the correct configuration to host the zone.

Because there are several ways to make the `zonepath` available on the new host, the actual movement of the `zonepath` from one system to another is a manual process that is performed by the global administrator.

When attached to the new system, the zone is in the installed state.

▼ How to Migrate A Non-Global Zone

You must be the global administrator in the global zone to perform this procedure.

1  **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in *System Administration Guide: Basic Administration*.

2  **Halt the zone to be migrated, my-zone in this procedure.**

   ```
   host1# zoneadm -z my-zone halt
   ```
3  **Detach the zone.**
host1# zoneadm -z my-zone detach
The detached zone is now in the configured state.

4  **Move the zonepath for my-zone to the new host.**
See “How to Move the zonepath to a new Host” on page 309 for more information.

5  **On the new host, configure the zone.**
host2# zonecfg -z my-zone
You will see the following system message:

my-zone: No such zone configured
Use 'create' to begin configuring a new zone.

6  **To create the zone my-zone on the new host, use the zonecfg command with the -a option and the zonepath on the new host.**
zonecfg:my-zone> create -a /export/zones/my-zone

7  **(Optional) View the configuration.**
zonecfg:my-zone> info
zonename: my-zone
zonepath: /export/zones/my-zone
autoboot: false
pool:
    inherit-pkg-dir:
        dir: /lib
    inherit-pkg-dir:
        dir: /platform
    inherit-pkg-dir:
        dir: /sbin
    inherit-pkg-dir:
        dir: /usr
net:
    address: 192.168.0.90
    physical: bge0

8  **Make any required adjustments to the configuration.**
For example, the network physical device is different on the new host, or devices that are part of
the configuration might have different names on the new host.
zonecfg:my-zone> select net physical=bge0
zonecfg:my-zone:net> set physical=e1000g0
zonecfg:my-zone:net> end
9  Commit the configuration and exit.
    zonecfg:my-zone> commit
    zonecfg:my-zone> exit

10  Attach the zone on the new host using one of the following methods.
    ■  Attach the zone with a validation check.
        host2# zoneadm -z my-zone attach
        The system administrator is notified of required actions to be taken if either or both of the following conditions are present:
        ■  Required packages and patches are not present on the new machine.
        ■  The software levels are different between machines.
    ■  Attach the zone with a validation check and update the zone to match a host running later versions of the dependent packages or having a different machine class upon attach.
        host2# zoneadm -z my-zone attach -u

        **Tip** – If the source system is running an older version of the Solaris system, it might not generate a correct list of packages when the zone is detached. To ensure that the correct package list is generated on the destination, you can remove the SUNWdetached.xml file from the zonepath. Removing this file will cause a new package list to be generated by the destination system.

    ■  Also use the -b option to back out specified patches, either official or IDR, before the update.
        host2# zoneadm -z my-zone attach -u -b IDR246802-01 -b 123456-08
        Note that you can use the -b option independently of the -u option.

    ■  Force the attach operation without performing the validation.
        host2# zoneadm -z my-zone attach -F

        **Caution** – The -F option allows you to force the attach with no validation performed. This is useful in certain cases, such as in a clustered environment or for backup and restore operations, but it does require that the system be properly configured to host the zone. An incorrect configuration could result in undefined behavior later.

**How to Move the zonepath to a new Host**

There are many ways to create an archive of the zonepath. For example, you can use the cpio or pax commands described in the `cpio(1)` and `pax(1)` man pages.
There are also several ways to transfer the archive to the new host. The mechanism used to transfer the zonepath from the source host to the destination depends on the local configuration. In some cases, such as a SAN, the zonepath data might not actually move. The SAN might simply be reconfigured so the zonepath is visible on the new host. In other cases, the zonepath might be written to tape, and the tape mailed to a new site.

For these reasons, this step is not automated. The system administrator must choose the most appropriate technique to move the zonepath to the new host.

1. **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. **Move the zonepath to the new host.** You can use the method described in this procedure, or use another method of your choice.

**Example 22–1 Archiving and Moving the zonepath Using the tar Command**

1. Create a tar file of the zonepath on host1 and transfer it to host2 by using the sftp command.

   ```bash
   host1# cd /export/zones
   host1# tar cf my-zone.tar my-zone
   host1# sftp host2
   Connecting to host2...
   Password: 
   sftp> cd /export/zones
   sftp> put my-zone.tar
   Uploading my-zone.tar to /export/zones/my-zone.tar
   sftp> quit
   ```

2. On host2, unpack the tar file.

   ```bash
   host2# cd /export/zones
   host2# tar xf my-zone.tar
   ```

For more information, see `sftp(1)` and `tar(1)`.

**Troubleshooting**

See “Resolving Problems With a zoneadm attach Operation” on page 407 for troubleshooting information on the following:

- Patches and packages are out of sync.
- Operating system releases do not match.

**Next Steps**

If you have copied the data instead of reconfiguring a SAN, then the zonepath data will still be visible on the source host even though the zone is now in the configured state. You can either
manually remove the zonepath from the source host after you have finished moving the data to
the new host, or you can reattach the zone to the source host and use the `zoneadm uninstall`
command to remove the zonepath.

**About Validating a Zone Migration Before the Migration Is Performed**

You can perform a trial run before the zone is moved to the new machine by using the "no
eexecute" option, `-n`.

The `zoneadm detach` subcommand is used with the `-n` option to generate a manifest on a
running zone without actually detaching the zone. The state of the zone on the originating
system is not changed. The zone manifest is sent to `stdout`. The global administrator can direct
this output to a file or pipe it to a remote command to be immediately validated on the target
host. The `zoneadm attach` subcommand is used with the `-n` option to read this manifest and
verify that the target machine has the correct configuration to host the zone without actually
doing an attach.

The zone on the target system does not have to be configured on the new host before doing a
trial-run attach.

**▼ How to Validate a Zone Migration Before the Migration Is Performed**

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With
   RBAC (Task Map)" in System Administration Guide: Basic Administration.

2. **Use one of the following methods.**
   - **Generate the manifest on a source host named `my - zone` and pipe the output to a remote
     command that will immediately validate the target host:**
     ```
     global# zoneadm -z my-zone detach -n | ssh remotehost zoneadm attach -n -
     ```
     The hyphen (`-`) at the end of the line specifies stdin for the path.
   - **Generate the manifest on a source host named `my - zone` and direct the output to a file:**
     ```
     global# zoneadm -z my-zone detach -n
     ```
Copy the manifest to the new host system as described in “How to Move the zonepath to a new Host” on page 309, and perform the validation:

```
global# zoneadm attach -n path_to_manifest
```
The path can be – to specify stdin.

## Migrating a Zone From a Machine That Is not Usable

A machine that hosts a native Solaris zone can become unusable. However, if the storage the zone lives on, such as a SAN, is still usable, it might still be possible to migrate the zone to a new host successfully. You can move the zonepath for the zone to the new host. In some cases, such as a SAN, the zonepath data might not actually move. The SAN might simply be re-configured so the zonepath is visible on the new host. Since the zone was not properly detached, you will have to first create the zone on the new host using the `zonecfg` command. Once this has been done, attach the zone on the new host. Although the new host will tell you the zone was not properly detached, the system will attempt the attach anyway.

The procedure for this task is described in steps 4 through 8 of “How to Migrate A Non-Global Zone” on page 307. Also see “How to Move the zonepath to a new Host” on page 309.
SX Only: Migrating a Physical Solaris System Into a Zone (Tasks)

A "physical to virtual" (P2V) capability is used to directly migrate an existing Solaris system into a native zone on a target system. This feature is not currently available on OpenSolaris 2009.06 systems.

Assess the System To Be Migrated

The system image to be installed through P2V must not be newer than the target host operating system release, or the installation will fail.

Depending on the services performed by the original system, the global administrator might need to manually customize the zone after it has been installed. For example, the privileges assigned to the zone might need to be modified. This is not done automatically. Also, because all system services do not work inside zones, not every physical system is a good candidate for migration into a zone.

To begin, examine the source system and collect needed information.

- Obtain the hostname:

  
  # hostname

- Obtain the host ID:

  
  # hostid

  Also see "Host ID Emulation" on page 315.

- Obtain the RPC domainname:

  
  # domainname

- Obtain the root password.

- View the software being run on the system:
Creating the Image for Directly Migrating A Solaris System Into a Zone

You can use the Flash Archiving tools to create an image of an installed system that can be migrated into a zone.

The image can be fully configured with all of the software that will be run in the zone. This image is used by the installer when the zone is installed.

▼ How to Use flarcreate to Create the Image

Use this process to create the system image. This example procedure uses NFS to place the flash archive on the target Solaris system, but you could use any method to move the files.

You must be the global administrator in the global zone to perform this procedure.

1. Become superuser, or assume the Primary Administrator role.

2. Log into the source system to be archived.

3. Change directories to the root directory.

   # cd /

4. Use flarcreate to create a flash archive image file named s-system, and place the archive onto the target system:

   target-system # flarcreate -S -n s-system /net/target/export/s-system.flar

   Determining which filesystems will be included in the archive...

   Creating the archive...

   cpio: File size of "etc/mnttab" has
increased by 435
2068650 blocks
1 error(s)
Archive creation complete.

Tip – In some cases, flarccreate can display errors from the cpio command. Most commonly, these are messages such as file size of etc/mnttab has increased by 33. When these messages pertain to log files or files that reflect system state, they can be ignored. Be sure to review all error messages thoroughly.

Other Archive Creation Methods

You can use alternate methods for creating the archive. The installer can accept the following archive formats:

■ cpio archives
■ gzip compressed cpio archives
■ bzip2 compressed cpio archives
■ pax archives created with the -x xustar (XUSTAR) format
■ ufsdump level zero (full) backups

Additionally, the installer can accept a directory of files created by using an archiving utility that saves and restores file permissions, ownership, and links. Thus, an example of a utility that cannot be used is tar, because tar does not handle links.

For more information, see the cpio(1), pax(1), bzip2(1), gzip(1), and ufsdump(1M) man pages.

Host ID Emulation

When applications are migrated from a standalone Solaris system into a zone on a new system, the hostid changes to be the hostid of the new machine.

In some cases, applications depend on the original hostid, and it is not possible to update the application configuration. In these cases, the zone can be configured to use the hostid of the original system. This is done by setting a zonecfg property to specify the hostid, as described in “How to Configure the Zone” on page 249. The value used should be the output of the hostid command as run on the original system. To view the hostid in an installed zone, also use the hostid command.

For more information about host IDs, see hostid(1).
Configure the Source Zone

Create the new zone configuration on the target system by using the procedure “How to Configure the Zone” on page 249.

Tip – If you know you will be using CDs or DVDs to install applications in the new zone, use add fs to add read-only access to CD or DVD media in the global zone when you initially configure the branded zone. A CD or DVD can then be used to install a product in the branded zone. See “How to Add Access to CD or DVD Media in a Non-Global Zone” on page 390 for more information.

Install the Zone

The zoneadm command described earlier in this guide and in the zoneadm(1M) man page is the primary tool used to install and administer non-global zones. Operations using the zoneadm command must be run from the global zone on the target system.

In addition to unpacking files from the archive, the install process performs checks, required postprocessing, and other functions to ensure that the zone is optimized to run on the host.

You can use an image of a Solaris system that has been fully configured with all of the software that will be run in the zone. See “Creating the Image for Directly Migrating A Solaris System Into a Zone” on page 314.

If you created a Solaris system archive from an existing system and use the -p (preserve sys1dcfg) option when you install the zone, then the zone will have the same identity as the system used to create the image.

If you use the -u (sys-unconfig) option when you install the target zone, the zone produced will not have a hostname or name service configured.

Caution – You must use either the -p option or the -u option. If you do not specify one of these two options, an error results.

Installer Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>Location of archive from which to copy system image. Full flash archive and cpio, gzip compressed cpio, bzip compressed cpio, and level 0 ufsdump are supported. Refer to the gzip man page available in the SUNWsfman package.</td>
</tr>
</tbody>
</table>
**Option** | **Description**
--- | ---
-d path | Location of directory from which to copy system image.
-d | Use the -d option with the dash parameter to direct that the existing directory layout be used in the zonepath. Thus, if the administrator manually sets up the zonepath directory before the installation, the -d option can be used to indicate that the directory already exists.
-p | Preserve system identity.
-s | Install silently.
-u sys-unconfig the zone.
-v | Verbose output.

The -a and -d options are mutually exclusive. The -p, -s, -u and -v options are only allowed when either -a or -d is provided.

**How to Install the Zone**

1. **Become superuser, or assume the Primary Administrator role.**

2. **Install the configured zone s-zone by using the zoneadm command with the install -a option and the path to the archive.**

   ```
global# zoneadm -z s-zone install -u -a /net/machine_name/s-system.flar
   ```

   You will see various messages as the installation completes. This can take some time.

   When the installation completes, use the list subcommand with the -i and -v options to list the installed zones and verify the status.

**Troubleshooting**

If an installation fails, review the log file. On success, the log file is in two places: /var/tmp in the global zone, and /var/log inside the zone. On failure, the log file is in /var/tmp.

If a zone installation is interrupted or fails, the zone is left in the incomplete state. Use uninstall -F to reset the zone to the configured state.
Boot the Zone

How to Boot the Zone

You must be the global administrator in the global zone to perform this procedure.

1  Become superuser, or assume the Primary Administrator role.

2  Use the `zoneadm` command with the `-z` option, the name of the zone, which is `s-zone`, and the `boot` subcommand to boot the zone.
   ```
global# zoneadm -z s-zone boot
   ```

3  When the boot completes, use the `list` subcommand with the `-v` option to verify the status.
   ```
global# zoneadm list -v
   ```
Both IPS and SVR4 packages are supported for the OpenSolaris 2009.06 release. This chapter discusses maintaining the Solaris Operating System on a system using SVR4 packaging when zones are installed.

Information about adding packages and patches to the operating system using SVR4 packaging in the global zone and in all installed non-global zones is provided. Information about removing packages and patches is also included. The material in this chapter supplements the existing Solaris installation and patch documentation. See the Solaris Express Release and Installation Collection and System Administration Guide: Basic Administration for more information.

This chapter covers the following SVR4 packaging topics:

- "SVR4 Packaging and Patch Tools Overview" on page 320
- "About SVR4 Packages and Zones" on page 321
- "Keeping Zones in Sync With SVR4 Packaging" on page 322
- "About Adding Packages in Zones (SVR4 Only)" on page 324
- "About Removing Packages in Zones (SVR4 Only)" on page 327
- "Package Parameter Information (SVR4 Only)" on page 328
- "Package Information Query" on page 336
- "About Adding Patches in Zones (SVR4 Only)" on page 336
- "Applying Patches on a Solaris System With Zones Installed (SVR4 Only)" on page 337
- "Removing Patches on a Solaris System With Zones Installed (SVR4 Only)" on page 339
- "PatchPro Support (SVR4 Only)" on page 339
- "Product Database (SVR4 Only)" on page 339
SVR4 Packaging and Patch Tools Overview

The Solaris packaging tools are used in administering the zones environment. The global administrator can upgrade the system to a new version of Solaris, which updates both the global and the non-global zones.

Solaris Live Upgrade, the standard Solaris interactive installation program, or the custom Solaris JumpStart installation program can be used in the global zone to upgrade a system that includes non-global zones.

The zone administrator can use the packaging tools to administer any software installed in a non-global zone, within the limits described in this document.

The following general principles apply when zones are installed:

- The global administrator can administer the software on every zone on the system.
- The root file system for a non-global zone can be administered from the global zone by using the Solaris packaging and patch tools. The Solaris packaging and patch tools are supported within the non-global zone for administering co-packaged (bundled), standalone (unbundled), or third-party products.
- The packaging and patch tools work in a zones-enabled environment. The tools allow a package or patch installed in the global zone to also be installed in a non-global zone.
- The SUNW_PKG_ALLZONES package parameter defines the zone scope of a package. The scope determines the type of zone in which an individual package can be installed. For more information about this parameter, see "SUNW_PKG_ALLZONES Package Parameter" on page 332.
- The SUNW_PKG_HOLLOW package parameter defines the visibility of a package if that package is required to be installed on all zones and be identical in all zones. For information about this parameter, see "SUNW_PKG_HOLLOW Package Parameter" on page 334.
- The SUNW_PKG_THISZONE package parameter defines whether a package must be installed in the current zone only. For information about this parameter, see "SUNW_PKG_THISZONE Package Parameter" on page 335.
- Packages that do not define values for zone package parameters have a default setting of false.
The packaging information visible from within a non-global zone is consistent with the files that have been installed in that zone using the Solaris packaging and patch tools. The visibility includes packages that have been imported from the global zone using read-only loopback mounts. See "Configuring, Verifying, and Committing a Zone" on page 249 for more information about this process.

A change, such as a patch or package added in the global zone, can be pushed out to all of the zones. This feature maintains consistency between the global zone and each non-global zone.

The package commands can add, remove, and interrogate packages. The patch commands can add and remove patches.

Note – While certain package and patch operations are performed, a zone is temporarily locked to other operations of this type. The system might also confirm a requested operation with the administrator before proceeding.

About SVR4 Packages and Zones

Only a subset of the Solaris packages installed on the global zone are completely replicated when a non-global zone is installed. For example, many packages that contain the Solaris kernel are not needed in a non-global zone. All non-global zones implicitly share the same Solaris kernel from the global zone. However, even if a package's data is not required or is not of use in a non-global zone, the knowledge that a package is installed in the global zone might be required in a non-global zone. The information allows package dependencies from the non-global zones to be properly resolved with the global zone.

Packages have parameters that control how their content is distributed and made visible on a system with non-global zones installed. The SUNW_PKG_ALLZONES, SUNW_PKG_HOLLOW, and SUNW_PKG_THISZONE package parameters define the characteristics of packages on a system with zones installed. If desired, system administrators can check these package parameter settings to verify the package's applicability when applying or removing a package in a zone environment. The pkgparam command can be used to view the values for these parameters. For more information on parameters, see "Package Parameter Information (SVR4 Only)" on page 328. See "Checking Package Parameter Settings on a System with Zones Installed" on page 349 for usage instructions.

Patches Generated for Packages

When a patch is generated for any package, the parameters must be set to the same values as the original package.
Interactive Packages

Any package that must be interactive, which means that it has a request script, is added to the current zone only. The package is not propagated to any other zone. If an interactive package is added to the global zone, the package is treated as though it is being added by using the pkgadd command with the -6 option. For more information about this option, see “About Adding Packages in Zones (SVR4 Only)” on page 324.

Keeping Zones in Sync With SVR4 Packaging

It is best to keep the software installed in the non-global zones in sync with the software installed in the global zone to the maximum extent possible. This practice minimizes the difficulty in administering a system with multiple installed zones.

To achieve this goal, the package tools enforce the following rules when adding or removing packages in the global zone.

Package Operations Possible in the Global Zone

If the package is not currently installed in the global zone and not currently installed in any non-global zone, the package can be installed:

- Only in the global zone, if SUNW_PKG_ALLZONES=false
- In the current zone only, which is the global zone in this case, if SUNW_PKG_THISZONE=true
- In the global zone and all non-global zones

If the package is currently installed in the global zone only:

- The package can be installed in all non-global zones.
- The package can be removed from the global zone.

If a package is currently installed in the global zone and currently installed in only a subset of the non-global zones:

- SUNW_PKG_ALLZONES must be set to false.
- The package can be installed in all non-global zones. Existing instances in any non-global zone are updated to the revision being installed.
- The package can be removed from the global zone.
- The package can be removed from the global zone and from all non-global zones.

If a package is currently installed in the global zone and currently installed in all non-global zones, the package can be removed from the global zone and from all non-global zones.

These rules ensure the following:
- Packages installed in the global zone are either installed in the global zone only, or installed in the global zone and all non-global zones.
- Packages installed in the global zone and also installed in any non-global zone are the same across all zones.

**Package Operations Possible in a Non-Global Zone**

The package operations possible in any non-global zone are:

- If a package is not currently installed in the non-global zone, the package can be installed only if `SUNW_PKG_ALLZONES=false`.
- The package can be installed in the current zone, which is the non-global zone in this case, if `SUNW_PKG_THISZONE=true`.
- If a package is currently installed in the non-global zone:
  - The package can be installed over the existing instance of the package only if `SUNW_PKG_ALLZONES=false`.
  - The package can be removed from the non-global zone only if `SUNW_PKG_ALLZONES=false`.

**How Zone State Affects Patch and Package Operations With SVR4 Packaging**

The following table describes what will happen when `pkgadd`, `pkgrm`, `patchadd`, and `patchrm` commands are used on a system with non-global zones in various states.

<table>
<thead>
<tr>
<th>Zone State</th>
<th>Effect on Package and Patch Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configured</td>
<td>Patch and package tools can be run. No software has been installed yet.</td>
</tr>
</tbody>
</table>
Zone State | Effect on Package and Patch Operations
--- | ---
Installed | Patch and package tools can be run. During patch or packaging operations, the system moves a zone from the installed state to a new internal state called mounted. After patching has completed, the zone is reverted back to the installed state.

Note that immediately after `zoneadm -z zonename install` has completed, the zone is also moved to the installed state. A zone in the installed state that has never been booted cannot be patched or run packaging commands. The zone must be booted to the running state at least once. After a zone has been booted at least once, and then moved back to the installed state by using `zoneadm halt`, then patch and packaging commands can be run.

Ready | Patch and package tools can be run.

Running | Patch and package tools can be run.

Incomplete | A zone being installed or removed by `zoneadm`. Patch and package tools cannot be used. The tools cannot bring the zone into the appropriate state for using the tools.

About Adding Packages in Zones (SVR4 Only)

The `pkgadd` system utility described in the `pkgadd(1M)` man page is used to add packages on a Solaris system with zones installed.

On the OpenSolaris 2009.06 release, use the `pkginstall` command.

Using `pkgadd` in the Global Zone

The `pkgadd` utility can be used with the `-G` option in the global zone to add the package to the global zone only. The package is not propagated to any other zones. Note that if `SUNW_PKG_THISZONE=true`, you do not have to use the `-G` option. If `SUNW_PKG_THISZONE=false`, the `-G` option will override it.

When you run the `pkgadd` utility in the global zone, the following actions apply.

- The `pkgadd` utility is able to add a package:
  - To the global zone only, unless the package is `SUNW_PKG_ALLZONES=true`
  - To the global zone and to all non-global zones
  - To all non-global zones only, if the package is already installed in the global zone
To the current zone only, if SUNW_PKG_THISZONE=true

The pkgadd utility cannot add a package:
- To any subset of the non-global zones
- To all non-global zones, unless the package is already installed in the global zone
- If the pkgadd utility is run without the -G option and SUNW_PKG_THISZONE=false, the specified package is added to all zones by default. The package is not marked as installed in the global zone only.
- If the pkgadd utility is run without the -G option and SUNW_PKG_THISZONE=true, then the specified package is added to the current (global) zone by default. The package is marked as installed in the global zone only.
- If the -G option is used, the pkgadd utility adds the specified package to the global zone only. The package is marked as installed in the global zone only. The package is not installed when any non-global zone is installed.

Adding a Package to the Global Zone and to All Non-Global Zones

To add a package to the global zone and to all non-global zones, execute the pkgadd utility in the global zone. As the global administrator, run pkgadd without the -G option.

A package can be added to the global zone and to all non-global zones without regard to the area affected by the package.

The following steps are performed by the pkgadd utility:
- Package dependencies are checked on the global zone and on all non-global zones. If required packages are not installed in any zone, then the dependency check fails. The system notifies the global administrator, who is prompted whether to continue.
- The package is added to the global zone.
- The package database on the global zone is updated.
- The package is added to each non-global zone and the database in the global zone is updated.
- The package database on each non-global zone is updated.

Adding a Package to the Global Zone Only

To add a package to the global zone only, as the global administrator in the global zone, execute the pkgadd utility with the -G option only.

A package can be added to the global zone if the following conditions are true:
- The package contents do not affect any area of the global zone that is shared with any non-global zone.
- The package is set SUNW_PKG_ALLZONES=false.
The following steps are performed by the `pkgadd` utility:

- If the package contents affect any area of the global zone that is shared with any non-global zone, or if the package is set `SUNW_PKG_ALLZONES=true`, then `pkgadd` fails. The error message states that the package must be added to the global zone and to all non-global zones.
- Package dependencies are checked on the global zone only. If required packages are not installed, then the dependency check fails. The system notifies the global administrator, who is prompted whether to continue.
- The package is added to the global zone.
- The package database on the global zone is updated.
- The package information on the global zone is annotated to indicate that this package is installed on the global zone only. If a non-global zone is installed in the future, this package will not be installed.

**Adding a Package Installed in the Global Zone to all Non-Global Zones**

To add a package that is already installed in the global zone to all non-global zones, you must currently remove the package from the global zone and reinstall it in all zones.

These are the steps used to add a package that is already installed in the global zone to all of the non-global zones:

1. In the global zone, use `pkgrm` to remove the package.
2. Add the package without using the `-G` option.

**Using `pkgadd` in a Non-Global Zone**

To add a package in a specified non-global zone, execute the `pkgadd` utility, without options, as the zone administrator. The following conditions apply:

- The `pkgadd` utility can only add packages in the non-global zone in which the utility is used.
- The package cannot affect any area of the zone that is shared from the global zone.
- The package must be set `SUNW_PKG_ALLZONES=false`.

The following steps are performed by the `pkgadd` utility:

- Package dependencies are checked on the non-global zone's package database before the package is added. If required packages are not installed, then the dependency check fails. The system notifies the non-global zone administrator, who is prompted whether to continue. The check fails if either of the following conditions are true.
  - Any component of the package affects any area of the zone that is shared from the global zone.
  - The package is set `SUNW_PKG_ALLZONES=true`. 
The package is added to the zone.
- The package database on the zone is updated.

About Removing Packages in Zones (SVR4 Only)

The `pkgrm` utility described in the `pkgrm(1M)` man page supports removing packages on a Solaris system with zones installed.

On the OpenSolaris 2009.06 release, use the `pkguninstall` command.

Using `pkgrm` in the Global Zone

The `pkgrm` utility can be used with the `-G` option from the global zone to remove packages from the global zone only. The package must not affect any area of the global zone shared with non-global zones or be installed in any non-global zone.

When the `pkgrm` utility is used in the global zone, the following actions apply:
- `pkgrm` can remove a package from the global zone and from all non-global zones, or from the global zone only when the package is only installed in the global zone.
- `pkgrm` cannot remove a package only from the global zone if the package is also installed in a non-global zone, or remove a package from any subset of the non-global zones.

Note that a package can only be removed from a non-global zone by a zone administrator working in that zone if the following are true:
- The package does not affect any area on the non-global zone that is shared from the global zone.
- The package is set `SUNW_PKG_ALLZONES=false`.

Removing a Package From the Global Zone and From all Non-Global Zones

To remove a package from the global zone and from all non-global zones, execute the `pkgrm` utility in the global zone. As the global administrator, run `pkgrm` without the `-G` option.

A package can be removed from the global zone and from all non-global zones without regard to the area affected by the package.

The following steps are performed by the `pkgrm` utility:
- Package dependencies are checked on the global zone and on all non-global zones. If the dependency check fails, then `pkgrm` fails. The system notifies the global administrator, who is prompted whether to continue.
The package is removed from each non-global zone.
The package database on each non-global zone is updated.
The package is removed from the global zone.
The package database on the global zone is updated.

Using `pkg rm` in a Non-Global Zone

As the zone administrator, use the `pkg rm` utility in a non-global zone to remove a package. The following limitations apply:

- `pkg rm` can only remove packages from the non-global zone.
- The `-G` option cannot be used. If this option is used, `pkg rm` outputs an error message and the attempted operation fails.
- The package cannot affect any area of the zone that is shared from the global zone.
- The package must be set `SUNW_PKG_ALLZONES=false`.

The following steps are performed by the `pkg rm` utility:

- Dependencies are checked on the non-global zone’s package database. If the dependency check fails, then `pkg rm` fails and the zone administrator is notified. The check fails if either of the following conditions are true.
  - Any component of the package affects any area of the zone that is shared from the global zone.
  - The package is set `SUNW_PKG_ALLZONES=true`.
- The package is removed from the zone.
- The package database on the zone is updated.

Package Parameter Information (SVR4 Only)

Setting Package Parameters for Zones

The `SUNW_PKG_ALLZONES`, `SUNW_PKG_HOLLOW`, and `SUNW_PKG_THISZONE` package parameters define the characteristics of packages on a system with zones installed. These parameters must be set so that packages can be administered on a system with non-global zones installed.

The following table lists the four valid combinations for setting package parameters. If you choose setting combinations that are not listed in the following table, those settings are invalid and the package will fail to install.
Ensure that you have set all three package parameters. You can leave all three package parameters blank. The package tools interpret a missing zone package parameter as if the setting were false, but not setting the parameters is strongly discouraged. By setting all three package parameters, you specify the exact behavior the package tools should exhibit when installing or removing the package.

**TABLE 24–1** Valid Package Parameter Settings

<table>
<thead>
<tr>
<th>SUNW_PKG_ALLZONES Setting</th>
<th>SUNW_PKG_HOLLOW Setting</th>
<th>SUNW_PKG_THISZONE Setting</th>
<th>Package Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>This is the default setting for packages that do not specify values for all the zone package parameters. A package with these settings can be installed in either the global zone or a non-global zone.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ If the pkgadd command is run in the global zone, the package is installed in the global zone and in all non-global zones.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ If the pkgadd command is run in a non-global zone, the package is installed in the non-global zone only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>true</td>
<td>In both cases, the entire contents of the package is visible in all zones where the package is installed.</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>A package with these settings can be installed in either the global zone or a non-global zone. If new non-global zones are created after the installation, the package is not propagated to these new non-global zones.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ If the pkgadd command is run in the global zone, the package is installed in the global zone only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>■ If the pkgadd command is run in a non-global zone, the package is installed in the non-global zone only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>In both cases, the entire contents of the package is visible in the zone where the package is installed.</td>
</tr>
<tr>
<td>Package Description</td>
<td>SUNW_PKG_ALLZONES</td>
<td>SUNW_PKG_HOLLOW</td>
<td>SUNW_PKG_THISZONE</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------</td>
<td>----------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>true false false</td>
<td>A package with these settings can be installed in the global zone only. When the <code>pkgadd</code> command is run, the package is installed in the global zone and in all non-global zones. The entire contents of the package is visible in all zones.</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUNW_PKG_ALLZONES Setting</td>
<td>SUNW_PKG_HOLLOW Setting</td>
<td>SUNW_PKG_THISZONE Setting</td>
<td>Package Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------</td>
<td>--------------------------</td>
<td>---------------------</td>
</tr>
</tbody>
</table>
| true                      | true                    | false                    | A package with these settings can only be installed in the global zone, by the global administrator. When the `pkgadd` command is run, the contents of the package is fully installed in the global zone. If a package has the package parameters set to these values, the package content itself is not delivered on any non-global zone. Only the package installation information necessary to make the package appear to be installed is installed on all non-global zones. This enables the installation of other packages to be installed that depend on this package. For package dependency checking purposes, the package appears to be installed in all zones.  
- In the global zone, the entire contents of the package is visible.  
- In whole root non-global zones, the entire contents of the package is not visible.  
- When a non-global zone inherits a file system from the global zone, a package installed in this file system is visible in a non-global zone. All other files delivered by the package are not visible within the non-global zone. For example, a native sparse root non-global zone shares certain directories with the global zone. These directories are read-only. Sparse root non-global zones share the `/platform` file system among others. Another example is packages that deliver files relevant only to booting hardware. |

**Note** – Any attempt to install the package in a non-global zone fails.
The optional SUNW_PKG_ALLZONES package parameter describes the zone scope of a package. This parameter defines the following:

- Whether a package is required to be installed on all zones
- Whether a package is required to be identical in all zones

The SUNW_PKG_ALLZONES package parameter has two permissible values. These values are true and false. The default value is false. If this parameter is either not set or set to a value other than true or false, the value false is used.

The SUNW_PKG_ALLZONES parameter should be set to true for packages that must be the same package version and patch revision level across all zones. Any package that delivers functionality dependent on a particular Solaris kernel, for example, Solaris 10, should set this parameter to true. Any patch for a package must set the SUNW_PKG_ALLZONES parameter to the same value that is set in the installed package being patched. The patch revision level for any package that sets this parameter to true must be the same across all zones.

Packages that deliver functionality not dependent on a particular Solaris kernel, such as third-party packages or Sun compilers, should set this parameter to false. Any patch for a package that sets this parameter to false must also set this parameter to false. Both the package version or the patch revision level for any package that sets this parameter to false can be different between zones. For example, two non-global zones could each have a different version of a web server installed.

The SUNW_PKG_ALLZONES package parameter values are described in the following table.
# TABLE 24–2  SUNW_PKG_ALLZONES Package Parameter Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| **false** | This package can be installed from the global zone to the global zone only, or to the global zone and to all non-global zones. The package can also be installed from any non-global zone to the same non-global zone.  
- The global administrator can install the package on the global zone only.  
- The global administrator can install the package on the global zone and on all non-global zones.  
- The zone administrator can install the package on a non-global zone.  
If removed from the global zone, the package is not removed from other zones. The package can be removed from individual non-global zones.  
- The package is not required to be installed on the global zone.  
- The package is not required to be installed on any non-global zone.  
- The package is not required to be identical across all zones. Different versions of the package can exist on individual zones.  
- The package delivers software that is not implicitly shared across all zones. This means that the package is not operating system-specific. Most application-level software is in this category. Examples include the StarOffice™ product or a web server. |
| **true** | If installed on the global zone, this package must also be installed on all non-global zones. If removed from the global zone, the package must also be removed from all non-global zones.  
- If the package is installed, it must be installed on the global zone. The package is then automatically installed on all non-global zones.  
- The version of the package must be identical on all zones.  
- The package delivers software that is implicitly shared across all zones. The package is dependent on the versions of software that are implicitly shared across all zones. The package should be visible in all non-global zones. Examples include kernel modules. These packages allow the non-global zone to resolve dependencies on packages that are installed in the global zone by requiring that the entire package be installed on all non-global zones.  
- Only the global administrator can install the package. A zone administrator cannot install the package on a non-global zone. |
**SUNW_PKG_HOLLOW** Package Parameter

The SUNW_PKG_HOLLOW package parameter defines whether a package should be visible in any non-global zone if that package is required to be installed and be identical in all zones.

The SUNW_PKG_HOLLOW package parameter has two permissible values, true or false.

- If SUNW_PKG_HOLLOW is either not set or set to a value other than true or false, the value false is used.
- If SUNW_PKG_ALLZONES is set to false, the SUNW_PKG_HOLLOW parameter is ignored.
- If SUNW_PKG_ALLZONES is set to false, then SUNW_PKG_HOLLOW cannot be set to true.

The SUNW_PKG_HOLLOW package parameter values are described in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| false  | This is not a “hollow” package:  
  - If installed on the global zone, the package content and installation information are required on all non-global zones.  
  - The package delivers software that should be visible in all non-global zones. An example is the package that delivers the truss command.  
  - Other than the restrictions for the current setting of the SUNW_PKG_ALLZONES package parameter, no additional restrictions are defined. |
SUNW_PKG_HOLLOW Package Parameter Values (Continued)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| true  | This is a "hollow" package:  
  ■ The package content is not delivered on any non-global zone. However, the package installation information is required on all non-global zones.  
  ■ The package delivers software that should not be visible in all non-global zones. Examples include kernel drivers and system configuration files that work only in the global zone. This setting allows the non-global zone to resolve dependencies on packages that are installed only on the global zone without actually installing the package data.  
  ■ The package is recognized as being installed in all zones for purposes of dependency checking by other packages that rely on this package being installed.  
  ■ This package setting includes all of the restrictions defined for setting SUNW_PKG_ALLZONES to true.  
  ■ In the global zone, the package is recognized as having been installed, and all components of the package are installed. Directories are created, files are installed, and class action and other scripts are run as appropriate when the package is installed.  
  ■ In a non-global zone, the package is recognized as having been installed, but no components of the package are installed. No directories are created, no files are installed, and no class action or other install scripts are run when the package is installed.  
  ■ When the package is removed from the global zone, the system recognizes that the package was completely installed. Appropriate directories and files are removed, and class action or other install scripts are run when the package is removed. |

SUNW_PKG_THISZONE Package Parameter

The SUNW_PKG_THISZONE package parameter defines whether a package must be installed in the current zone, global or non-global, only. The SUNW_PKG_THISZONE package parameter has two permissible values. These values are true and false. The default value is false.

The SUNW_PKG_THISZONE package parameter values are described in the following table.
### Table 24-4  SUNW_PKG_THISZONE Package Parameter Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>- If <code>pkgadd</code> is run in a non-global zone, the package is installed in the current zone only.</td>
</tr>
<tr>
<td></td>
<td>- If <code>pkgadd</code> is run in the global zone, the package is installed in the global zone and also</td>
</tr>
<tr>
<td></td>
<td>installed in all currently installed non-global zones. In addition, the package will be</td>
</tr>
<tr>
<td></td>
<td>propagated to all future, newly installed non-global zones.</td>
</tr>
<tr>
<td>true</td>
<td>- The package is installed in the current zone only.</td>
</tr>
<tr>
<td></td>
<td>- If installed in the global zone, the package is not added to any currently existing or</td>
</tr>
<tr>
<td></td>
<td>yet-to-be-created non-global zones. This is the same behavior that occurs when the -g</td>
</tr>
<tr>
<td></td>
<td>option is specified to <code>pkgadd</code>.</td>
</tr>
</tbody>
</table>

---

**Package Information Query**

The `pkginfo` utility described in the `pkginfo(1)` man page supports querying the software package database on a Solaris system with zones installed. For information about the database, see “Product Database (SVr4 Only)” on page 339.

The `pkginfo` utility can be used in the global zone to query the software package database in the global zone only. The `pkginfo` utility can be used in a non-global zone to query the software package database in the non-global global zone only.

On the OpenSolaris 2009.06 release, use the `pkginfo` command.

**About Adding Patches in Zones (SVR4 Only)**

In general, a patch consists of the following components:

- **Patch information:**
  - Identification, which is the patch version and patch ID
  - Applicability, which is the operating system type, operating system version, and architecture
  - Dependencies, such as requires and obsoletes
  - Properties, such as requires a reboot afterwards
- **One or more packages to patch,** where each package contains:
  - The version of the package to which the patches can be applied
  - Patch information, such as ID, obsoletes, and requires
One or more components of the package to be patched

When the `patchadd` command is used to apply a patch, the patch information is used to determine whether the patch is applicable to the currently running system. If determined to be not applicable, the patch is not applied. Patch dependencies are also checked against all of the zones on the system. If any required dependencies are not met, the patch is not applied. This could include the case in which a later version of the patch is already installed.

Each package contained in the patch is checked. If the package is not installed on any zone, then the package is bypassed and not patched.

If all dependencies are satisfied, all packages in the patch that are installed on any zone are used to patch the system. The package and patch databases are also updated.

**Applying Patches on a Solaris System With Zones Installed (SVr4 Only)**

All patches applied at the global zone level are applied across all zones. When a non-global zone is installed, it is at the same patch level as the global zone. When the global zone is patched, all non-global zones are similarly patched. This action maintains the same patch level across all zones.

The `patchadd` system utility described in the `patchadd(1M)` man page is used to add patches on a system with zones installed.

**Using patchadd in the Global Zone**

To add a patch to the global zone and to all non-global zones, run `patchadd` as the global administrator in the global zone.

When `patchadd` is used in the global zone, the following conditions apply:

- The `patchadd` utility is able to add the patch(es) to the global zone and to all non-global zones only. This is the default action.
- The `patchadd` utility cannot add the patch(es) to the global zone only or to a subset of the non-global zones.

When you add a patch to the global zone and to all non-global zones, you do not have to consider whether the patch affects areas that are shared from the global zone.

The following steps are performed by the `patchadd` utility:

- The patch is added to the global zone.
- The patch database on the global zone is updated.
The patch is added to each non-global zone.
- The patch database on each non-global zone is updated.

**Using patchadd in a Non-Global Zone**

When used in a non-global zone by the zone administrator, *patchadd* can only be used to add patches to that zone. A patch can be added to a non-global zone in the following cases:
- The patch does not affect any area of the zone that is shared from the global zone.
- All packages in the patch are set \texttt{SUNW_PKG_ALLZONES=false}.

The following steps are performed by the *patchadd* utility:
- The patch is added to the zone.
- The patch database on the zone is updated.

**Interaction of patchadd -G and the pkginfo Variable on a System With Zones**

The following list specifies the interaction between the \texttt{-G} option and the \texttt{SUNW_PKG_ALLZONES} variable when adding a patch in global and non-global zones.

**Global zone, \texttt{-G} specified**

- If any packages have \texttt{SUNW_PKG_ALLZONES=TRUE}, this results in an error and no action.
- If no packages have \texttt{SUNW_PKG_ALLZONES=TRUE}, patch is applied to package(s) in global zone only.

**Global zone, \texttt{-G} not specified**

- If any packages have \texttt{SUNW_PKG_ALLZONES=TRUE}, patch is applied to those package(s) in all zones.
- If any packages do not have \texttt{SUNW_PKG_ALLZONES=TRUE}, patch is applied to those package(s) in all appropriate zones. Global zone only packages are installed only in the global zone.

**Non-global zone, \texttt{-G} specified or not specified**

- If any packages have \texttt{SUNW_PKG_ALLZONES=TRUE}, this results in an error and no action.
If no packages have `SUNW_PKG_ALLZONES=TRUE`, patch is applied to packages in non-global zone only.

Removing Patches on a Solaris System With Zones Installed (SVR4 Only)

The `patchrm` system utility described in the `patchrm(1M)` man page is used to remove patches on a system with zones installed.

Using `patchrm` in the Global Zone

As the global administrator, you can use the `patchrm` utility in the global zone to remove patches. The `patchrm` utility cannot remove patches from the global zone only or from a subset of the non-global zones.

Using `patchrm` in a Non-Global Zone

As the zone administrator, you can use the `patchrm` utility in a non-global zone to remove patches from that non-global zone only. Patches cannot affect areas that are shared.

PatchPro Support (SVR4 Only)

PatchPro can be used in the global zone and in any non-global zone. If run in the global zone, PatchPro uses the existing patch database and patch tools to patch the global and all non-global zones for all software that is installed on the global zone. No software installed in a non-global zone that is not also installed in the global zone will be taken into account.

A zone administrator can run PatchPro in a non-global zone to patch the software installed in the non-global zone.

Product Database (SVR4 Only)

Each zone's respective package, patch, and product registry database completely describes all installed software that is available on the zone. All dependency checking for installing additional software or patches is performed without accessing any other zone's database, unless
a package or patch is being installed or removed on the global zone and on one or more non-global zones. In this case, the appropriate non-global zone database(s) must be accessed.

For more information about the database, see the pkgadm(1M) man page.
Adding and Removing Packages and Patches on a Solaris System With Zones Installed (Tasks)

This chapter describes how to add and remove packages and patches on a system using SVR4 packaging with zones installed. Other tasks associated with managing packages and patches, such as checking package parameter settings and obtaining package information, are also addressed. For an overview of patching and packaging concepts on a system using SVR4 packaging with zones installed, see Chapter 24, "About Packages and Patches on a Solaris System With Zones Installed (Overview)."

To learn about IPS packaging on an OpenSolaris 2009.06 system, see OpenSolaris 2009.06 Image Packaging System Guide (http://dlc.sun.com/osol/docs/content/2009.06/IMGPACKAGESYS/).

Adding and Removing Packages and Patches on a Solaris System With Zones Installed (Task Map)

Applicable to SVR4 only:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a package.</td>
<td>Add a package on a system with zones installed.</td>
<td>&quot;Adding a Package on a Solaris System With Zones Installed&quot; on page 342</td>
</tr>
<tr>
<td>Check package information.</td>
<td>Check package information on a system with zones installed.</td>
<td>&quot;Checking Package Information on a Solaris System With Zones Installed&quot; on page 345</td>
</tr>
<tr>
<td>Remove a package.</td>
<td>Remove a package on a system with zones installed.</td>
<td>&quot;Removing a Package From a Solaris System With Zones Installed&quot; on page 346</td>
</tr>
</tbody>
</table>
Task | Description | For Instructions
--- | --- | ---
Apply a patch. | Apply a patch on a system with zones installed. | “Applying a Patch to a Solaris System With Zones Installed” on page 347
Remove a patch. | Remove a patch on a system with zones installed. | “Removing a Patch on a System with Zones Installed” on page 348
(Optional) Check the package parameter settings. | When adding or removing packages, verify that the settings of the package parameters support the action you want to perform. | “Checking Package Parameter Settings on a System with Zones Installed” on page 349

### Adding a Package on a Solaris System With Zones Installed

You can use the `pkgadd` system utility described in the `pkgadd(1M)` man page to perform the following tasks:

- Add a package to the global zone only
- Add a package to both the global zone and all non-global zones
- Add a package that is already installed in the global zone to the non-global zones
- Add a package to a specified non-global zone only

The `SUNW_PKG_ALLZONES` and `SUNW_PKG_HOLLOW` package parameter settings must match the correct value, either `true` or `false`, to add packages. Otherwise, the desired result will not be achieved. For more information about the effect of these package parameter settings, see “About SVR4 Packages and Zones” on page 321. For more information about how to check these package parameter settings, see “Checking Package Parameter Settings on a System with Zones Installed” on page 349.

#### ▼ How to Add a Package to the Global Zone Only

To add a package to the global zone only, the `SUNW_PKG_ALLZONES` package parameter must be set to `false`.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**

   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **While in the global zone, run the `pkgadd -d` command followed by the location of the package, the `-G` option, and then the package name.**
If installing the package from a CD-ROM, type:

```
global# pkgadd -d /cdrom/cdrom0/directory -G package_name
```

If installing the package from a directory to which it has been copied, type:

```
global# pkgadd -d disk1/image -G package_name
```

where `disk1` is the location where the package was copied.

**Note** – If the `pkgadd` utility is run without the `-G` option and `SUNW_PKG_THISZONE=true`, then the specified package is added to the current (global) zone by default.

▼ **How to Add a Package to the Global Zone and All Non-Global Zones**

Do not use `pkgadd` option `-G` in this procedure.

You must be the global administrator in the global zone to perform this procedure.

1 **Become superuser, or assume the Primary Administrator role.**

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2 **While in the global zone, run the `pkgadd -d` command followed by the location of the package and then the package name.**

   - If installing the package from a CD-ROM, type:

```
global# pkgadd -d /cdrom/cdrom0/directory package_name
```

   - If installing the package from a directory to which it has been copied, type:

```
global# pkgadd -d disk1/image package_name
```

   where `disk1` is the location where the package was copied.
How to Add a Package That Is Installed in the Global Zone to All Non-Global Zones

You must be the global administrator in the global zone to perform this procedure.

1. Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2. In the global zone, use pkg rm to remove the package.

3. Add the package without using the -G option.

How to Add a Package to a Specified Non-Global Zone Only

To add a package to a specified non-global zone only, the SUNW_PKG_ALLZONES package parameter must be set to false. Do not use the pkgadd option -G in this procedure or the operation fails.

You must be the zone administrator in the non-global zone to perform this procedure.

1. Log in to the non-global zone as the zone administrator.

2. While in the non-global zone, my-zone in this procedure, run the pkgadd -d command followed by the location of the package and then the package name.
   - If installing the package from a CD-ROM, type:
     
     ```
     my-zone# pkgadd -d /cdrom/cdrom0/directory package_name
     ```
   - If installing the package from a directory to which it has been copied, type:
     
     ```
     my-zone# pkgadd -d disk1/image package_name
     ```
     where disk1 is the location where the package was copied.
Checking Package Information on a Solaris System With Zones Installed

You can query the software package database for the global zone and non-global zones by using the `pkginfo` command. See the `pkginfo(1)` man page for more information about this command.

▼ How to Check Package Information in the Global Zone Only

- To check the software package database for the global zone only, use `pkginfo` followed by the package name.

```
global% pkginfo package_name
```

**Example 25–1** Using the `pkginfo` Command in the Global Zone

```
global% pkginfo SUNWcsr SUNWcsu
system SUNWcsr Core Solaris, (Root)
system SUNWcsu Core Solaris, (Usr)
```

▼ How to Check Package Information in a Specified Non-Global Zone Only

- To check the software package database in a specific non-global zone, log into the non-global zone and use `pkginfo` followed by the package name.

```
my-zone% pkginfo package_name
```

**Example 25–2** Using the `pkginfo` Command in a Non-Global Zone

```
my-zone% pkginfo SUNWcsr SUNWcsu
system SUNWcsr Core Solaris, (Root)
system SUNWcsu Core Solaris, (Usr)
```
Removing a Package From a Solaris System With Zones Installed

You can use the `pkg rm` system utility described in the `pkg rm(1M)` man page to perform the following tasks:

- Remove a package from the global zone and all non-global zones
- Remove a package from a specified non-global zone only

The `SUNW_PKG_ALLZONES` and `SUNW_PKG_HOLLOW` package parameter settings must match the correct value, either true or false, to remove packages. Otherwise, the desired result will not be achieved. For more information about the effect of these package parameter settings, see “About SVR4 Packages and Zones” on page 321. For more information about how to check these package parameter settings, see “Checking Package Parameter Settings on a System with Zones Installed” on page 349.

▼ How to Remove a Package From the Global Zone and All Non-Global Zones

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**

   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. **While in the global zone, run the `pkg rm` command followed by the package name.**

   ```
   global# pkg rm package_name
   ```

▼ How to Remove a Package From a Specified Non-Global Zone Only

To remove a package from a specified non-global zone only, the `SUNW_PKG_ALLZONES` package parameter must be set to `false`.

You must be the zone administrator in the non-global zone to perform this procedure.

1. **Log in to the non-global zone as the zone administrator.**

2. **While in the non-global zone, `my-zone` in this procedure, run the `pkg rm` command followed by the package name.**

   ```
   my-zone# pkg rm package_name
   ```
Applying a Patch to a Solaris System With Zones Installed

You can use the `patchadd` system utility described in the `patchadd(1M)` man page to perform the following tasks:

- Apply a patch to the global zone only
- Apply a patch to the global zone and all non-global zones
- Apply a patch to specified non-global zone only

▼ How to Apply a Patch to the Global Zone Only

You must be the global administrator in the global zone to perform this procedure.

1 **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 **Execute the `patchadd` command with the `-G` option, followed by the patch ID.**
   ```
   global# patchadd -G patch_id
   ```

▼ How to Apply a Patch to the Global Zone and All Non-Global Zones

You must be the global administrator in the global zone to perform this procedure.

1 **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 **Execute the `patchadd` command followed by the patch ID.**
   ```
   global# patchadd patch_id
   ```

▼ How to Apply a Patch to a Specified Non-Global Zone Only

To apply a patch to a specified non-global zone only, the `SUNW_PKG_ALLZONES` package parameter for all packages in the patch set must be set to `false`.

You must be the zone administrator in the non-global zone to perform this procedure.

1 **Log in to the non-global zone as the zone administrator.**
Removing a Patch on a System with Zones Installed

2. While in the non-global zone, my-zone in this procedure, execute the `patchadd` command followed by the patch ID.

```
my-zone# patchadd patch_id
```

### Removing a Patch on a System with Zones Installed

You can use the `patchrm` system utility described in the `patchrm(1M)` man page to perform the following task:

- Remove a patch from the global zone and all non-global zones
- Remove a patch from a specified non-global zone only

#### ▼ How to Remove a Patch From the Global Zone and All Non-Global Zones

You must be the global administrator in the global zone to perform this procedure.

1. Become superuser, or assume the Primary Administrator role.
   
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in *System Administration Guide: Basic Administration*.

2. Execute the `patchrm` command followed by the patch ID.

```
global# patchrm patch_id
```

#### ▼ How to Remove a Patch From a Specified Non-Global Zone Only

To remove a patch from a specified non-global zone only, the SUNW_PKG_ALLZONES package parameter for all packages in the patch set must be set to false.

You must be the zone administrator in the non-global zone to perform this procedure.

1. Log in to the non-global zone as the zone administrator.

2. While in the non-global zone, my-zone in this procedure, execute the `patchrm` command followed by the patch ID.

```
my-zone# patchrm patch_id
```
Checking Package Parameter Settings on a System with Zones Installed

Before you add or remove a software package, you can use the `pkgparam` command to check package parameter settings. This step is optional. This check also can be done when troubleshooting why a package is not added or removed as expected. For information about displaying package parameter values, see the `pkgparam(1)` man page.

**▼ (Optional) How to Check the Setting of a Package Already Installed on the System**

- To check the package parameter setting of a package that is already installed in a global or non-global zone, use `pkgparam` followed by the package name and the name of the parameter.

  ```
  my-zone% pkgparam package_name SUNW_PKG_ALLZONES
  true
  my-zone% pkgparam package_name SUNW_PKG_HOLLOW
  false
  ```

**▼ (Optional) How to Check the Setting of a Package in Software on a CD-ROM**

- To check the package parameter setting of an uninstalled package in software located on a CD-ROM, use `pkgparam -d` with the path of the CD-ROM followed by the package name and the name of the parameter.

  ```
  my-zone% pkgparam -d /cdrom/cdrom0/directory package_name SUNW_PKG_ALLZONES
  true
  my-zone% pkgparam -d /cdrom/cdrom0/directory package_name SUNW_PKG_HOLLOW
  false
  ```
This chapter covers these general zone administration topics:

- “Global Zone Visibility and Access” on page 352
- “Process ID Visibility in Zones” on page 352
- “System Observability in Zones” on page 352
- “Non-Global Zone Node Name” on page 353
- “File Systems and Non-Global Zones” on page 353
- “Networking in Shared-IP Non-Global Zones” on page 360
- “Networking in Exclusive-IP Non-Global Zones” on page 362
- “Device Use in Non-Global Zones” on page 364
- “Running Applications in Non-Global Zones” on page 365
- “Resource Controls Used in Non-Global Zones” on page 366
- “Fair Share Scheduler on a Solaris System With Zones Installed” on page 366
- “Extended Accounting on a Solaris System With Zones Installed” on page 367
- “Privileges in a Non-Global Zone” on page 367
- “Using IP Security Architecture in Zones” on page 371
- “Using Solaris Auditing in Zones” on page 372
- “Core Files in Zones” on page 373
- “Running DTrace in a Non-Global Zone” on page 373
- “About Backing Up a Solaris System With Zones Installed” on page 374
- “Determining What to Back Up in Non-Global Zones” on page 375
- “Commands Used on a Solaris System With Zones Installed” on page 377

For information on lx branded zones, see Part III.
Global Zone Visibility and Access

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

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

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

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

Process ID Visibility in Zones

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

System Observability in Zones

The `ps` command has the following modifications:

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

For more information, see the ps(1) man page.

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

- `ipcs` (see the `ipcs(1)` man page)
- `pgrep` (see the `pgrep(1)` man page)
- `p tree` (see the `proc(1)` man page)
- `prstat` (see the `prstat(1M)` man page)

See Table 26–5 for the full list of changes made to commands.

### Non-Global Zone Node Name

The node name in `/etc/nodename` returned by `uname -n` can be set by the zone administrator. The node name must be unique.

### File Systems and Non-Global Zones

This section provides information about file system issues on a Solaris system with zones installed. Each zone has its own section of the file system hierarchy, rooted at a directory known as the zone root. Processes in the zone can access only files in the part of the hierarchy that is located under the zone root. The `chroot` utility can be used in a zone, but only to restrict the process to a root path within the zone. For more information about `chroot`, see `chroot(1M)`.

### The -onosuid Option

The -o nosuid option to the `mount` utility has the following functionality:

- Processes from a `setuid` binary located on a file system that is mounted using the nosetuid option do not run with the privileges of the `setuid` binary. The processes run with the privileges of the user that executes the binary.
  
  For example, if a user executes a `setuid` binary that is owned by `root`, the processes run with the privileges of the user.

- Opening device-special entries in the file system is not allowed. This behavior is equivalent to specifying the `nodevices` option.

This file system-specific option is available to all Solaris file systems that can be mounted with `mount` utilities, as described in the `mount(1M)` man page. In this guide, these file systems are...

Mounting File Systems in Zones

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

Options for mounting file systems in non-global zones are described in the following table. Procedures for these mounting alternatives are provided in “Configuring, Verifying, and Committing a Zone” on page 249 and “Mounting File Systems in Running Non-Global Zones” on page 387.

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.

<table>
<thead>
<tr>
<th>File System</th>
<th>Mounting Options in a Non-Global Zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoFS</td>
<td>Cannot be mounted using zonecfg, cannot be manually mounted from the global zone into a non-global zone. Can be mounted from within the zone.</td>
</tr>
<tr>
<td>CacheFS</td>
<td>Cannot be used in a non-global zone.</td>
</tr>
<tr>
<td>FDFS</td>
<td>Can be mounted using zonecfg, can be manually mounted from the global zone into a non-global zone, can be mounted from within the zone.</td>
</tr>
<tr>
<td>HSFS</td>
<td>Can be mounted using zonecfg, can be manually mounted from the global zone into a non-global zone, can be mounted from within the zone.</td>
</tr>
<tr>
<td>LOFS</td>
<td>Can be mounted using zonecfg, can be manually mounted from the global zone into a non-global zone, can be mounted from within the zone.</td>
</tr>
<tr>
<td>MNTFS</td>
<td>Cannot be mounted using zonecfg, cannot be manually mounted from the global zone into a non-global zone. Can be mounted from within the zone.</td>
</tr>
</tbody>
</table>
File Systems and Non-Global Zones

<table>
<thead>
<tr>
<th>File System</th>
<th>Mounting Options in a Non-Global Zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFS</td>
<td>Cannot be mounted using zonecfg. V2, V3, and V4, which are the versions currently supported in zones, can be mounted from within the zone.</td>
</tr>
<tr>
<td>PCFS</td>
<td>Can be mounted using zonecfg, can be manually mounted from the global zone into a non-global zone, can be mounted from within the zone.</td>
</tr>
<tr>
<td>PROCFS</td>
<td>Cannot be mounted using zonecfg, cannot be manually mounted from the global zone into a non-global zone. Can be mounted from within the zone.</td>
</tr>
<tr>
<td>TMPFS</td>
<td>Can be mounted using zonecfg, can be manually mounted from the global zone into a non-global zone, can be mounted from within the zone.</td>
</tr>
<tr>
<td>UDFS</td>
<td>Can be mounted using zonecfg, can be manually mounted from the global zone into a non-global zone, can be mounted from within the zone.</td>
</tr>
<tr>
<td>UFS</td>
<td>Can be mounted using zonecfg, can be manually mounted from the global zone into a non-global zone, can be mounted from within the zone.</td>
</tr>
<tr>
<td>XMEMFS</td>
<td>Support for this file system has been removed from Solaris with this release.</td>
</tr>
<tr>
<td>ZFS</td>
<td>Can be mounted using the zonecfg dataset and fs resource types.</td>
</tr>
</tbody>
</table>

For more information, see “How to Configure the Zone” on page 249, “Mounting File Systems in Running Non-Global Zones” on page 387, and the `mount(1M)` man page.

**Unmounting File Systems in Zones**

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

There are security restrictions on mounting certain file systems from within a zone. Other file systems exhibit special behavior when mounted in a zone. The list of modified file systems follows.

AutoFS

AutoFS is a client-side service that automatically mounts the appropriate file system. When a client attempts to access a file system that is not presently mounted, the AutoFS file system intercepts the request and calls automountd to mount the requested directory. AutoFS mounts established within a zone are local to that zone. The mounts cannot be accessed from other zones, including the global zone. The mounts are removed when the zone is halted or rebooted. For more information on AutoFS, see “How Autos Works” in System Administration Guide: Network Services.

Each zone runs its own copy of automountd. The auto maps and timeouts are controlled by the zone administrator. You cannot trigger a mount in another zone by crossing an AutoFS mount point for a non-global zone from the global zone.

Certain AutoFS mounts are created in the kernel when another mount is triggered. Such mounts cannot be removed by using the regular umount interface because they must be mounted or unmounted as a group. Note that this functionality is provided for zone shutdown.

MNTFS

MNTFS is a virtual file system that provides read-only access to the table of mounted file systems for the local system. The set of file systems visible by using mnt tab from within a non-global zone is the set of file systems mounted in the zone, plus an entry for root (/). Mount points with a special device that is not accessible from within the zone, such as /dev/rdsk/c0t0d0s0, have their special device set to the same as the mount point. All mounts in the system are visible from the global zone’s /etc/mnt tab table. For more information on MNTFS, see Chapter 19, “Mounting and Unmounting File Systems (Tasks),” in System Administration Guide: Devices and File Systems.

NFS

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

As documented in the mount_nfs(1M) man page, an NFS server should not attempt to mount its own file systems. Thus, a zone should not NFS mount a file system exported by the global zone. Zones cannot be NFS servers. From within a zone, NFS mounts behave as though mounted with the nodevices option.

The nfsstat command output only pertains to the zone in which the command is run. For example, if the command is run in the global zone, only information about the global zone is reported. For more information about the nfsstat command, see nfsstat(1M).
The `zlogin` command will fail if any of its open files or any portion of its address space reside on NFS. For more information, see “zlogin Command” on page 289.

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

Processes in the global zone can observe processes and other objects in non-global zones. This allows such processes to have system-wide observability.

From within a zone, procfs mounts behave as though mounted with the `nodevices` option. For more information about procfs, see the proc(4) man page.

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

UFS, UDFS, PCFS, and other storage-based file systems
When using the zonetfg command to configure storage-based file systems that have an fsck binary, such as UFS, the zone administrator must specify a raw parameter. The parameter indicates the raw (character) device, such as `/dev/rdsk/c0t0d0s7`. zoneadmd automatically runs the fsck command in non-interactive check-only mode (`fsck -m`) on this device before it mounts the file system. If the fsck fails, zoneadmd cannot bring the zone to the ready state. The path specified by raw cannot be a relative path.

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

For more information, see “The zoneadmd Daemon” on page 268 and the fsck(1M)

ZFS
You can add a ZFS dataset to a non-global zone by using the zonetfg command with the add dataset resource. The dataset will be visible and mounted in the non-global zone and no longer visible in the global zone. The zone administrator can create and destroy file systems within that dataset, and modify the properties of the dataset.

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

```
# zfs get zoned tank/sales
NAME PROPERTY VALUE SOURCE
  tank/sales    zoned  on  local
```

If you want to share a dataset from the global zone, you can add an LOFS-mounted ZFS file system by using the zonetfg command with the add fs subcommand. The global administrator is responsible for setting and controlling the properties of the dataset.
For more information on ZFS, see Chapter 10, “ZFS Advanced Topics,” in *Solaris ZFS Administration Guide*.

## Non-Global Zones as NFS Clients

Zones can be NFS clients. Version 2, version 3, and version 4 protocols are supported. For information on these NFS versions, see “Features of the NFS Service” in *System Administration Guide: Network Services*.

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

- You can edit `/etc/default/nfs` to set `NFS_CLIENT_VERSMAX=number` so that the zone uses the specified version by default. See “Setting Up NFS Services” in *System Administration Guide: Network Services*. Use the procedure How to Select Different Versions of NFS on a Client by Modifying the `/etc/default/nfs` File from the task map.

- You can manually create a version mount. This method overrides the contents of `/etc/default/nfs`. See “Setting Up NFS Services” in *System Administration Guide: Network Services*. Use the procedure How to Use the Command Line to Select Different Versions of NFS on a Client from the task map.

### Use of `mknod` Prohibited in a Zone

Note that you cannot use the `mknod` command documented in the `mknod(1M)` man page to make a special file in a non-global zone.

## Traversing File Systems

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

- Specifying that the zone root’s parent directory is owned, readable, writable, and executable by root only

- Restricting access to directories exported by `/proc`

Note that attempting to access AutoFS nodes mounted for another zone will fail. The global administrator must not have auto maps that descend into other zones.
Restriction on Accessing A Non-Global Zone From the Global Zone

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

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

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

An example is the -R root_path option to the pkgadd utility run from the global zone with a non-global zone root path.

The list of commands, programs, and utilities that use -R with an alternative root path include the following:

- auditreduce
- bart
- flar
- flarcreate
- installf
- localeadm
- makeuuid
- metaroot
- patchadd
- patchrm
- pkgadd
- pkqadm
- pkqask
- pkgchk
- pkgrm
- prodreg
- removef
- routeadm
- showrev
- syseventadm

The list of commands and programs that use -b with an alternative root path include the following:
Networking in Shared-IP Non-Global Zones

On a Solaris system with zones installed, the zones can communicate with each other over the network. The zones all have separate bindings, or connections, and the zones can all run their own server daemons. These daemons can listen on the same port numbers without any conflict. The IP stack resolves conflicts by considering the IP addresses for incoming connections. The IP addresses identify the zone.

Shared-IP Zone Partitioning

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

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

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

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

- Internet services daemon with a full configuration file (see the inetd(1M) man page)
- sendmail (see the sendmail(1M) man page)
- apache (see the apache(1M) man page)

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

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

Within a non-global zone, only that zone's interfaces will be visible to `ifconfig`.

For more information, see the `ifconfig(1M)` and `if_tcp(7P)` man pages.

IP Traffic Between Shared-IP Zones on the Same Machine

Between two zones on the same machine, packet delivery is only allowed if there is a “matching route” for the destination and the zone in the forwarding table.

The matching information is implemented as follows:

- The source address for the packets is selected on the output interface specified by the matching route.
- By default, traffic is permitted between two zones that have addresses on the same subnet. The matching route in this case is the interface route for the subnet.
- If there is a default route for a given zone, where the gateway is on one of the zone's subnets, traffic from that zone to all other zones is allowed. The matching route in this case is the default route.
- If there is a matching route with the RTF\_REJECT flag, packets trigger an ICMP unreachable message. If there is a matching route with the RTF\_BLACKHOLE flag, packets are discarded. The global administrator can use the `route` command options described in the following table to create routes with these flags.

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-reject</td>
<td>RTF_REJECT</td>
<td>Emit an ICMP unreachable message when matched.</td>
</tr>
<tr>
<td>-blackhole</td>
<td>RTF_BLACKHOLE</td>
<td>Silently discard packets during updates.</td>
</tr>
</tbody>
</table>

For more information, see the `route(1M)`
Solaris IP Filter in Shared-IP Zones

Solaris IP Filter provides stateful packet filtering and network address translation (NAT). A stateful packet filter can monitor the state of active connections and use the information obtained to determine which network packets to allow through the firewall. Solaris IP Filter also includes stateless packet filtering and the ability to create and manage address pools. See Chapter 25, "Solaris IP Filter (Overview),” in System Administration Guide: IP Services for additional information.

Solaris IP Filter can be enabled in non-global zones by turning on loopback filtering as described in Chapter 26, "Solaris IP Filter (Tasks),” in System Administration Guide: IP Services.

Solaris IP Filter is derived from open source IP Filter software.

IP Network Multipathing in Shared-IP Zones

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

All network configuration is done in the global zone. You can configure IPMP in the global zone, then extend the functionality to non-global zones. The functionality is extended by placing the zone’s address in an IPMP group when you configure the zone. Then, if one of the interfaces in the global zone fails, the non-global zone addresses will migrate to another network interface card.

In a given non-global zone, only the interfaces associated with the zone are visible through the ifconfig command.

See “How to Extend IP Network Multipathing Functionality to Shared-IP Non-Global Zones” on page 394. The zones configuration procedure is covered in “How to Configure the Zone” on page 249. For information on IPMP features, components, and usage, see Chapter 12, “Introducing IPMP” in System Administration Guide: Network Interfaces and Network Virtualization.

Networking in Exclusive-IP Non-Global Zones

An exclusive-IP zone has its own IP-related state and tuning variables. The zone is assigned its own set of data-links when the zone is configured.

For information on features that can be used in an exclusive-IP non-global zone, see “Exclusive-IP Non-Global Zones” on page 220. For information on tuning IP ndd variables, see Solaris Tunable Parameters Reference Manual.
Exclusive-IP Zone Partitioning

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

Exclusive-IP Data-Link Interfaces

A data-link name must be assigned exclusively to a single zone.

The `dladm show-link` command can be used to display data-links assigned to running zones.

For more information, see `dladm(1M)`

IP Traffic Between Exclusive-IP Zones on the Same Machine

There is no internal loopback of IP packets between exclusive-IP zones. All packets are sent down to the data-link. Typically, this means that the packets are sent out on a network interface. Then, devices like Ethernet switches or IP routers can forward the packets toward their destination, which might be a different zone on the same machine as the sender.

Solaris IP Filter in Exclusive-IP Zones

You have the same IP Filter functionality that you have in the global zone in an exclusive-IP zone. IP Filter is also configured the same way in exclusive-IP zones and the global zone.

IP Network Multipathing in Exclusive-IP Zones

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

The data-link configuration is done in the global zone. First, multiple data-link interfaces are assigned to a zone using `zonecfg`. The multiple data-link interfaces must be attached to the same IP subnet. IPMP can then be configured from within the exclusive-IP zone by the zone administrator.
Device Use in Non-Global Zones

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

/dev and the /devices Namespace

The devfs file system described in the devfs(7FS) man page is used by the Solaris system to manage /devices. Each element in this namespace represents the physical path to a hardware device, pseudo-device, or nexus device. The namespace is a reflection of the device tree. As such, the file system is populated by a hierarchy of directories and device special files.

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

The /dev file hierarchy is managed by the dev file system described in the dev(7FS) man page.

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

Exclusive-Use Devices

You might have devices that you want to assign to specific zones. Allowing unprivileged users to access block devices could permit those devices to be used to cause system panic, bus resets, or other adverse effects. Before making such assignments, consider the following issues:

- Before assigning a SCSI tape device to a specific zone, consult the sgen(7D) man page.
- Placing a physical device into more than one zone can create a covert channel between zones. Global zone applications that use such a device risk the possibility of compromised data or data corruption by a non-global zone.

Device Driver Administration

In a non-global zone, you can use the modinfo command described in the modinfo(1M) man page to examine the list of loaded kernel modules.
Most operations concerning kernel, device, and platform management will not work inside a non-global zone because modifying platform hardware configurations violates the zone security model. These operations include the following:

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

Utilities That Do Not Work or Are Modified in Non-Global Zones

Utilities That Do Not Work in Non-Global Zones

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

- `prtconf` (see the `prtconf(1M)` man page)
- `prtdiag` (see the `prtdiag(1M)` man page)

SPARC: Utility Modified for Use in a Non-Global Zone

The `eeprom` utility can be used in a zone to view settings. The utility cannot be used to change settings. For more information, see the `eeprom(1M)` and `openprom(7D)` man pages.

Running Applications in Non-Global Zones

In general, all applications can run in a non-global zone. However, the following types of applications might not be suitable for this environment:

- Applications that use privileged operations that affect the system as a whole. Examples include operations that set the global system clock or lock down physical memory.
- The few applications dependent upon certain devices that do not exist in a non-global zone, such as `/dev/kmem`.
- Applications that expect to be able to write into `/usr`, either at runtime or when being installed, patched, or upgraded. This is because `/usr` is read-only for a non-global zone by default. Sometimes the issues associated with this type of application can be mitigated without changing the application itself.
- In a shared-IP zone, applications dependent upon devices in `/dev/ip`.
Resource Controls Used in Non-Global Zones

For additional information about using a resource management feature in a zone, also refer to the chapter that describes the feature in Part 1 of this guide.

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

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

A “one zone, one pool” rule applies to non-global zones. Multiple non-global zones can share the resources of one pool. Processes in the global zone, however, can be bound by a sufficiently privileged process to any pool. The resource controller poold only runs in the global zone, where there is more than one pool for it to operate on. The poolstat utility run in a non-global zone displays only information about the pool associated with the zone. The pooladm command run without arguments in a non-global zone displays only information about the pool associated with the zone.

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

Fair Share Scheduler on a Solaris System With Zones Installed

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

FSS Share Division in a Global or Non-Global Zone

FSS CPU shares for a zone are hierarchical. The shares for the global and non-global zones are set by the global administrator through the zone-wide resource control zone.cpu-shares. The project.cpu-shares resource control can then be defined for each project within that zone to further subdivide the shares set through the zone-wide control.

To assign zone shares by using the zonecfg command, see “How to Set zone.cpu-shares in the Global Zone” on page 260. For more information on project.cpu-shares, see “Available Resource Controls” on page 86. Also see “Using the Fair Share Scheduler on a Solaris System With Zones Installed” on page 397 for example procedures that show how to set shares on a temporary basis.
Share Balance Between Zones

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

Extended Accounting on a Solaris System With Zones Installed

The extended accounting subsystem collects and reports information for the entire system (including non-global zones) when run in the global zone. The global administrator 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.


Privileges in a Non-Global Zone

Processes are restricted to a subset of privileges. Privilege restriction prevents a zone from performing operations that might affect other zones. The set of privileges limits the capabilities of privileged users within the zone. To display the list of privileges available from within a given zone, use the `ppriv` utility.

The following table lists all of the Solaris privileges and the status of each privilege with respect to zones. Optional privileges are not part of the default set of privileges but can be specified through the `limitpriv` property. Required privileges must be included in the resulting privilege set. Prohibited privileges cannot be included in the resulting privilege set.

**TABLE 26–1 Status of Privileges in Zones**

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Status</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpc_cpu</td>
<td>Optional</td>
<td>Access to certain cpc(3CPC) counters</td>
</tr>
<tr>
<td>dtrace_proc</td>
<td>Optional</td>
<td>fasttrap and pid providers; plockstat(1M)</td>
</tr>
<tr>
<td>Privilege</td>
<td>Status</td>
<td>Notes</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>dtrace_user</td>
<td>Optional</td>
<td>profile and syscall providers</td>
</tr>
<tr>
<td>graphics_access</td>
<td>Optional</td>
<td>ioctl(2) access to agpgart_io(7I)</td>
</tr>
<tr>
<td>graphics_map</td>
<td>Optional</td>
<td>mmap(2) access to agpgart_io(7I)</td>
</tr>
<tr>
<td>entication</td>
<td>Raw</td>
<td>PF_INET/PF_INET6 packet access</td>
</tr>
<tr>
<td>proc_clock_highres</td>
<td>Optional</td>
<td>Use of high resolution timers</td>
</tr>
<tr>
<td>proc_prioctnl</td>
<td>Optional</td>
<td>Scheduling control; priocntl(1)</td>
</tr>
<tr>
<td>sys_ipc_config</td>
<td>Optional</td>
<td>Raising IPC message queue buffer size</td>
</tr>
<tr>
<td>sys_time</td>
<td>Optional</td>
<td>System time manipulation; xntp(1M)</td>
</tr>
<tr>
<td>dtrace_kernel</td>
<td>Prohibited</td>
<td>Currently unsupported</td>
</tr>
<tr>
<td>proc_zone</td>
<td>Prohibited</td>
<td>Currently unsupported</td>
</tr>
<tr>
<td>sys_config</td>
<td>Prohibited</td>
<td>Currently unsupported</td>
</tr>
<tr>
<td>sys_devices</td>
<td>Prohibited</td>
<td>Currently unsupported</td>
</tr>
<tr>
<td>sys_linkdir</td>
<td>Prohibited</td>
<td>Currently unsupported</td>
</tr>
<tr>
<td>sys_net_config</td>
<td>Prohibited</td>
<td>Currently unsupported</td>
</tr>
<tr>
<td>sys_res_config</td>
<td>Prohibited</td>
<td>Currently unsupported</td>
</tr>
<tr>
<td>sys_suser_compat</td>
<td>Prohibited</td>
<td>Currently unsupported</td>
</tr>
<tr>
<td>proc_exec</td>
<td>Required, Default</td>
<td>Used to start init(1M)</td>
</tr>
<tr>
<td>proc_fork</td>
<td>Required, Default</td>
<td>Used to start init(1M)</td>
</tr>
<tr>
<td>sys_mount</td>
<td>Required, Default</td>
<td>Needed to mount required file systems</td>
</tr>
<tr>
<td>sys_ip_config</td>
<td>Required</td>
<td>Default in exclusive-IP zones</td>
</tr>
<tr>
<td></td>
<td>Required</td>
<td>in exclusive-IP zones</td>
</tr>
<tr>
<td></td>
<td>Required</td>
<td>Prohibited in shared-IP zones</td>
</tr>
<tr>
<td></td>
<td>Required</td>
<td>to boot zone and initialize IP networking in exclusive-IP zone</td>
</tr>
<tr>
<td>contract_event</td>
<td>Default</td>
<td>Used by contract file system</td>
</tr>
<tr>
<td>contract_observer</td>
<td>Default</td>
<td>Contract observation regardless of UID</td>
</tr>
<tr>
<td>file_chown</td>
<td>Default</td>
<td>File ownership changes</td>
</tr>
<tr>
<td>file_chown_self</td>
<td>Default</td>
<td>Owner/group changes for own files</td>
</tr>
<tr>
<td>file_dac_execute</td>
<td>Default</td>
<td>Execute access regardless of mode/ACL</td>
</tr>
</tbody>
</table>
### Privileges in a Non-Global Zone

**TABLE 26-1** Status of Privileges in Zones  *(Continued)*

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Status</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>file_dac_read</td>
<td>Default</td>
<td>Read access regardless of mode/ACL</td>
</tr>
<tr>
<td>file_dac_search</td>
<td>Default</td>
<td>Search access regardless of mode/ACL</td>
</tr>
<tr>
<td>file_dac_write</td>
<td>Default</td>
<td>Write access regardless of mode/ACL</td>
</tr>
<tr>
<td>file_link_any</td>
<td>Default</td>
<td>Link access regardless of owner</td>
</tr>
<tr>
<td>file_owner</td>
<td>Default</td>
<td>Other access regardless of owner</td>
</tr>
<tr>
<td>file_setid</td>
<td>Default</td>
<td>Permission changes for setid, setgid, setuid files</td>
</tr>
<tr>
<td>ipc_dac_read</td>
<td>Default</td>
<td>IPC read access regardless of mode</td>
</tr>
<tr>
<td>ipc_dac_owner</td>
<td>Default</td>
<td>IPC write access regardless of mode</td>
</tr>
<tr>
<td>ipc_owner</td>
<td>Default</td>
<td>IPC other access regardless of mode</td>
</tr>
<tr>
<td>net_icmpaccess</td>
<td>Default</td>
<td>ICMP packet access: ping(1M)</td>
</tr>
<tr>
<td>net_privaddr</td>
<td>Default</td>
<td>Binding to privileged ports</td>
</tr>
<tr>
<td>proc_audit</td>
<td>Default</td>
<td>Generation of audit records</td>
</tr>
<tr>
<td>proc_chroot</td>
<td>Default</td>
<td>Changing of root directory</td>
</tr>
<tr>
<td>proc_info</td>
<td>Default</td>
<td>Process examination</td>
</tr>
<tr>
<td>proc_lock_memory</td>
<td>Default</td>
<td>Locking memory; shmctl(2) and mlock(3C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If this privilege is assigned to a non-global zone by the system administrator, consider also setting the zone.max-locked-memory resource control to prevent the zone from locking all memory.</td>
</tr>
<tr>
<td>proc_owner</td>
<td>Default</td>
<td>Process control regardless of owner</td>
</tr>
<tr>
<td>proc_session</td>
<td>Default</td>
<td>Process control regardless of session</td>
</tr>
<tr>
<td>proc_setid</td>
<td>Default</td>
<td>Setting of user/group IDs at will</td>
</tr>
<tr>
<td>proc_taskid</td>
<td>Default</td>
<td>Assigning of task IDs to caller</td>
</tr>
<tr>
<td>sys_acct</td>
<td>Default</td>
<td>Management of accounting</td>
</tr>
<tr>
<td>sys_admin</td>
<td>Default</td>
<td>Simple system administration tasks</td>
</tr>
<tr>
<td>sys_audit</td>
<td>Default</td>
<td>Management of auditing</td>
</tr>
<tr>
<td>sys_nfs</td>
<td>Default</td>
<td>NFS client support</td>
</tr>
<tr>
<td>sys_resource</td>
<td>Default</td>
<td>Resource limit manipulation</td>
</tr>
</tbody>
</table>
The following table lists all of the Solaris Trusted Extensions privileges and the status of each privilege with respect to zones. Optional privileges are not part of the default set of privileges but can be specified through the `limitpriv` property.

**Note** – Trusted Solaris privileges are interpreted only if the system is configured with Trusted Extensions.

### TABLE 26-2 Status of Solaris Trusted Extensions Privileges in Zones

<table>
<thead>
<tr>
<th>Solaris Trusted Extensions Privilege</th>
<th>Status</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>file_downgrade_sl</td>
<td>Optional</td>
<td>Set the sensitivity label of file or directory to a sensitivity label that does not dominate the existing sensitivity label</td>
</tr>
<tr>
<td>file_upgrade_sl</td>
<td>Optional</td>
<td>Set the sensitivity label of file or directory to a sensitivity label that dominates the existing sensitivity label</td>
</tr>
<tr>
<td>sys_trans_label</td>
<td>Optional</td>
<td>Translate labels not dominated by sensitivity label</td>
</tr>
<tr>
<td>win_colormap</td>
<td>Optional</td>
<td>Colormap restrictions override</td>
</tr>
<tr>
<td>win_config</td>
<td>Optional</td>
<td>Configure or destroy resources that are permanently retained by the X server</td>
</tr>
<tr>
<td>win_dac_read</td>
<td>Optional</td>
<td>Read from window resource not owned by client’s user ID</td>
</tr>
<tr>
<td>win_dac_write</td>
<td>Optional</td>
<td>Write to or create window resource not owned by client’s user ID</td>
</tr>
<tr>
<td>win_devices</td>
<td>Optional</td>
<td>Perform operations on input devices.</td>
</tr>
<tr>
<td>win_dga</td>
<td>Optional</td>
<td>Use direct graphics access X protocol extensions; frame buffer privileges needed</td>
</tr>
<tr>
<td>win_downgrade_sl</td>
<td>Optional</td>
<td>Change sensitivity label of window resource to new label dominated by existing label</td>
</tr>
<tr>
<td>win_fontpath</td>
<td>Optional</td>
<td>Add an additional font path</td>
</tr>
<tr>
<td>win_mac_read</td>
<td>Optional</td>
<td>Read from window resource with a label that dominates the client’s label</td>
</tr>
<tr>
<td>win_mac_write</td>
<td>Optional</td>
<td>Write to window resource with a label not equal to the client’s label</td>
</tr>
<tr>
<td>win_selection</td>
<td>Optional</td>
<td>Request data moves without confirmor intervention</td>
</tr>
</tbody>
</table>
TABLE 26–2  Status of Solaris Trusted Extensions Privileges in Zones  (Continued)

<table>
<thead>
<tr>
<th>Solaris Trusted Extensions Privilege</th>
<th>Status</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>win_upgrade_sl</td>
<td>Optional</td>
<td>Change sensitivity label of window resource to a new label not dominated by existing label</td>
</tr>
<tr>
<td>net_bindmlp</td>
<td>Default</td>
<td>Allows binding to a multilevel port (MLP)</td>
</tr>
<tr>
<td>net_mac_aware</td>
<td>Default</td>
<td>Allows reading down through NFS</td>
</tr>
</tbody>
</table>

To alter privileges in a non-global zone configuration, see “Configuring, Verifying, and Committing a Zone” on page 249.

To inspect privilege sets, see “Using the ppriv Utility” on page 383. For more information about privileges, see the ppriv(1) man page and System Administration Guide: Security Services.

**Using IP Security Architecture in Zones**

The Internet Protocol Security Architecture (IPsec), which provides IP datagram protection, is described in Chapter 19, “IP Security Architecture (Overview),” in System Administration Guide: IP Services. The Internet Key Exchange (IKE) protocol is used to manage the required keying material for authentication and encryption automatically.

For more information, see the ipsecconf(1M) and ipseckey(1M) man pages.

**IP Security Architecture in Shared-IP Zones**

IPsec can be used in the global zone. However, IPsec in a non-global zone cannot use IKE. Therefore, you must manage the IPsec keys and policy for the non-global zones by using the Internet Key Exchange (IKE) protocol in the global zone. Use the source address that corresponds to the non-global zone that you are configuring.

**IP Security Architecture in Exclusive-IP Zones**

IPsec can be used in exclusive-IP zones.
Using Solaris Auditing in Zones

Solaris auditing is described in Chapter 28, “Solaris Auditing (Overview),” in System Administration Guide: Security Services. For zones considerations associated with auditing, see the following sections:

- “Auditing and Solaris Zones” in System Administration Guide: Security Services

An audit record describes an event, such as logging in to a system or writing to a file. The record is composed of tokens, which are sets of audit data. By using the zonename token, you can configure Solaris auditing to identify audit events by zone. Use of the zonename token allows you to produce the following information:

- Audit records that are marked with the name of the zone that generated the record
- An audit log for a specific zone that the global administrator can make available to the zone administrator

Configuring Audit in the Global Zone

Solaris audit trails are configured in the global zone. Audit policy is set in the global zone and applies to processes in all zones. The audit records can be marked with the name of the zone in which the event occurred. To include zone names in audit records, you must edit the /etc/security/audit_startup file before you install any non-global zones. The zone name selection is case-sensitive.

To configure auditing in the global zone to include all zone audit records, add this line to the /etc/security/audit_startup file:

```
/usr/sbin/auditconfig -setpolicy +zonename
```

As the global administrator in the global zone, execute the auditconfig utility:

```
global# auditconfig -setpolicy +zonename
```

For additional information, see the audit_startup(1M) and auditconfig(1M) man pages and “Configuring Audit Files (Task Map)” in System Administration Guide: Security Services.
### Configuring User Audit Characteristics in a Non-Global Zone

When a non-global zone is installed, the `audit_control` file and the `audit_user` file in the global zone are copied to the zone's `/etc/security` directory. These files might require modification to reflect the zone's audit needs.

For example, each zone can be configured to audit some users differently from others. To apply different per-user preselection criteria, both the `audit_control` and the `audit_user` files must be edited. The `audit_user` file in the non-global zone might also require revisions to reflect the user base for the zone if necessary. Because each zone can be configured differently with regard to auditing users, it is possible for the `audit_user` file to be empty.

For additional information, see the `audit_control(4)` and `audit_user(4)` man pages.

### Providing Audit Records for a Specific Non-Global Zone

By including the `zonename` token as described in “Configuring Audit in the Global Zone” on page 372, Solaris audit records can be categorized by zone. Records from different zones can then be collected by using the `auditreduce` command to create logs for a specific zone.

For more information, see the `audit_startup(1M)` and `auditreduce(1M)` man pages.

### Core Files in Zones

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

For more information, see the `coreadm(1M)` and `core(4)` man pages.

### Running DTrace in a Non-Global Zone

DTrace programs that only require the `dtrace_proc` and `dtrace_user` privileges can be run in a non-global zone. To add these privileges to the set of privileges available in the non-global zone, use the `zonecfg limitpriv` property. For instructions, see “How to Use DTrace” on page 385.

The providers supported through `dtrace_proc` are `fasttrap` and `pid`. The providers supported through `dtrace_user` are `profile` and `syscall`. DTrace providers and actions are limited in scope to the zone.
About Backing Up a Solaris System With Zones Installed

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

Backing Up Loopback File System Directories

Because many non-global zones share files with the global zone through the use of loopback file system read-only mounts (usually /usr, /lib, /sbin, and /platform), you must use a global zone backup method to back up lofs directories.

Caution – Do not back up the lofs file systems in non-global zones. An attempt by the non-global administrator to restore lofs file systems from a non-global zone could cause a serious problem.

Backing Up Your System From the Global Zone

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

- You want to back up the configurations of your non-global zones as well as the application data.
- Your primary concern is the ability to recover from a disaster. If you need to restore everything or almost everything on your system, including the root file systems of your zones and their configuration data as well as the data in your global zone, backups should take place in the global zone.
- You want to use the ufsdump command to perform a data backup. Because importing a physical disk device into a non-global zone would change the security profile of the zone, ufsdump should only be used from the global zone.
- You have commercial network backup software.

Note – Your network backup software should be configured to skip all inherited lofs file systems if possible. The backup should be performed when the zone and its applications have quiesced the data to be backed up.
Backing Up Individual Non-Global Zones on Your System

You might decide to perform backups within the non-global zones in the following cases.

- The non-global zone administrator needs the ability to recover from less serious failures or to restore application or user data specific to a zone.
- You want to use programs that back up on a file-by-file basis, such as tar or cpio. See the tar(1) and cpio(1) man pages.
- You use the backup software of a particular application or service running in a zone. It might be difficult to execute the backup software from the global zone because application environments, such as directory path and installed software, would be different between the global zone and the non-global zone.

If the application can perform a snapshot on its own backup schedule in each non-global zone and store those backups in a writable directory exported from the global zone, the global zone administrator can pick up those individual backups as part of the backup strategy from the global zone.

Determining What to Back Up in Non-Global Zones

You can back up everything in the non-global zone, or, because a zone's configuration changes less frequently, you can perform backups of the application data only.

Backing Up Application Data Only

If application data is kept in a particular part of the file system, you might decide to perform regular backups of this data only. The zone's root file system might not have to be backed up as often because it changes less frequently.

You will have to determine where the application places its files. Locations where files can be stored include the following:

- Users' home directories
- /etc for configuration data files
- /var

Assuming the application administrator knows where the data is stored, it might be possible to create a system in which a per-zone writable directory is made available to each zone. Each zone can then store its own backups, and the global administrator can make this location one of the places on the system to back up.
General Database Backup Operations

If the database application data is not under its own directory, the following rules apply:

- Ensure that the databases are in a consistent state first.
  Databases must be quiesced because they have internal buffers to flush to disk. Make sure that the databases in non-global zones have come down before starting the backup from the global zone.
- Within each zone, use file system features to make a snapshot of the data, then back up the snapshots directly from the global zone.
  This process will minimize elapsed time for the backup window and remove the need for backup clients/modules in all of the zones.

Tape Backups

Each non-global zone can take a snapshot of its private file systems when it is convenient for that zone and the application has been briefly quiesced. Later, the global zone can back up each of the snapshots and put them on tape after the application is back in service.

This method has the following advantages:

- Fewer tape devices are needed.
- There is no need for coordination between the non-global zones.
- There is no need to assign devices directly to zones, which improves security.
- Generally, this method keeps system management in the global zone, which is preferred.

About Restoring Non-Global Zones

In the case of a restore where the backups were done from the global zone, the global administrator can reinstall the affected zones and then restore that zone's files. Note that this assumes the following:

- The zone being restored has the same configuration as it did when the backup was done.
- The global zone has not been upgraded or patched 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.

For example, you might need to merge files by hand if a global zone has been patched after the backup, but prior to the restore of the non-global zone. In this case, you would have to be careful when restoring a zone's files that were backed up since a backed up file might not be compatible with the newly installed zone that was built after the patches were applied to the
global zone. In this case, you would have to examine the files individually and compare them to
the copies in the newly installed zone. In most cases, you will find that the file can be copied
directly in, but in some cases, you must merge the changes originally made to the file into the
newly installed or patched copy in the zone.

**Note** – If all file systems in the global zone are lost, restoring everything in the global zone
restores the non-global zones as well, as long as the respective root file systems of the non-global
zones were included in the backup.

## Commands Used on a Solaris System With Zones Installed

The commands identified in Table 26–3 provide the primary administrative interface to the
zones facility.

<table>
<thead>
<tr>
<th>Command Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>zlogin(1)</td>
<td>Log in to a non-global zone</td>
</tr>
<tr>
<td>zonename(1)</td>
<td>Prints the name of the current zone</td>
</tr>
<tr>
<td>zoneadm(1M)</td>
<td>Administers zones on a system</td>
</tr>
<tr>
<td>zonecfg(1M)</td>
<td>Used to set up a zone configuration</td>
</tr>
<tr>
<td>getzoneid(3C)</td>
<td>Used to map between zone ID and name</td>
</tr>
<tr>
<td>zones(5)</td>
<td>Provides description of zones facility</td>
</tr>
<tr>
<td>zcons(7D)</td>
<td>Zone console device driver</td>
</tr>
</tbody>
</table>

The `zoneadm` daemon is the primary process for managing the zone’s virtual platform. The
man page for the `zoneadm` daemon is `zoneadm(1M)`. The daemon does not constitute a
programming interface.

The commands in the next table are used with the resource capping daemon.

<table>
<thead>
<tr>
<th>Command Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rcapstat(1)</td>
<td>Monitors the resource utilization of capped projects.</td>
</tr>
<tr>
<td>rcapadm(1M)</td>
<td>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</td>
</tr>
</tbody>
</table>
The commands identified in the following table have been modified for use on a Solaris system with zones installed. These commands have options that are specific to zones or present information differently. The commands are listed by man page section.

### TABLE 26–4  Commands Used With rcapd  (Continued)

<table>
<thead>
<tr>
<th>Command Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rcapd(1M)</td>
<td>The resource capping daemon.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipcrm(1)</td>
<td>Added <code>-z zone</code> option. This option is only useful when the command is executed in the global zone.</td>
</tr>
<tr>
<td>ipcs(1)</td>
<td>Added <code>-z zone</code> option. This option is only useful when the command is executed in the global zone.</td>
</tr>
<tr>
<td>pgrrep(1)</td>
<td>Added <code>-z zoneidlist</code> option. This option is only useful when the command is executed in the global zone.</td>
</tr>
<tr>
<td>ppriv(1)</td>
<td>Added the expression <code>zone</code> for use with the <code>-l</code> option to list all privileges available in the current zone. Also use the option <code>-v</code> after <code>zone</code> to obtain verbose output.</td>
</tr>
<tr>
<td>priocntl(1)</td>
<td>Zone ID can be used in <code>idlist</code> and <code>-i idtype</code> to specify processes. You can use the <code>priocntl -i zoneid</code> command to move running processes into a different scheduling class in a non-global zone.</td>
</tr>
<tr>
<td>proc(1)</td>
<td>Added <code>-z zone</code> option to <code>ptree</code> only. This option is only useful when the command is executed in the global zone.</td>
</tr>
</tbody>
</table>
| ps(1)             | Added `zonename` and `zoneid` to list of recognized `format` names used with the `-o` option.  

Added `-z zonelist` to list only processes in the specified zones. Zones can be specified either by zone name or by zone ID. This option is only useful when the command is executed in the global zone.  

Added `-Z` to print the name of the zone associated with the process. The name is printed under an additional column header, `ZONE`. |
| renice(1)         | Added `zoneid` to list of valid arguments used with the `-i` option. |
| sar(1)            | If executed in a non-global zone in which the pools facility is enabled, the `-b`, `-c`, `-g`, `-m`, `-p`, `-u`, `-w`, and `-y` options display values only for processors that are in the processor set of the pool to which the zone is bound. |
| auditconfig(1M)   | Added `zonename` token. |
| auditreduce(1M)   | Added `-z zone-name` option. Added ability to get an audit log of a zone. |
### TABLE 26–5  Commands Modified for Use on a Solaris System With Zones Installed (Continued)

<table>
<thead>
<tr>
<th>Command Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>coreadm(1M)</code></td>
<td>Added variable <code>%z</code> to identify the zone in which process executed.</td>
</tr>
<tr>
<td><code>df(1M)</code></td>
<td>Added <code>-Z</code> option to display mounts in all visible zones. This option has no effect in a non-global zone.</td>
</tr>
<tr>
<td><code>ifconfig(1M)</code></td>
<td>Added <code>zone</code> option for global zone use (the default), and <code>-zone zonename</code> for non-global zone use.</td>
</tr>
<tr>
<td><code>iostat(1M)</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>kstat(1M)</code></td>
<td>If executed in the global zone, <code>kstat</code> are displayed for all zones. If executed in a non-global zone, only <code>kstat</code> with a matching <code>zoneid</code> are displayed.</td>
</tr>
<tr>
<td><code>mpstat(1M)</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>ndd(1M)</code></td>
<td>When used in the global zone, displays information for all zones. <code>ndd</code> on the TCP/IP modules in an exclusive-IP zone only displays information for that zone.</td>
</tr>
<tr>
<td><code>netstat(1M)</code></td>
<td>Displays information for the current zone only.</td>
</tr>
<tr>
<td><code>nfsstat(1M)</code></td>
<td>Displays statistics for the current zone only.</td>
</tr>
<tr>
<td><code>poolbind(1M)</code></td>
<td>Added <code>zoneid</code> list. Also see &quot;Resource Pools Used in Zones&quot; on page 145 for information about using zones with resource pools.</td>
</tr>
<tr>
<td><code>prstat(1M)</code></td>
<td>Added <code>-z zoneidlist</code> option. Also added <code>-Z</code> 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 <code>-a</code>, <code>-t</code>, <code>-T</code>, <code>-J</code>, and <code>-Z</code> options displays a SWAP instead of a SIZE column. The swap reported is the total swap consumed by the zone’s processes and <code>tmpfs</code> mounts. This value assists in monitoring the swap reserved by each zone, which can be used to choose a reasonable <code>zone.max-swap</code> setting.</td>
</tr>
<tr>
<td><code>psrinfo(1M)</code></td>
<td>If executed in a non-global zone, only information about the processors visible to the zone is displayed.</td>
</tr>
<tr>
<td><code>traceroute(1M)</code></td>
<td>Usage change. When specified from within a non-global zone, the <code>-F</code> option has no effect because the “don’t fragment” bit is always set.</td>
</tr>
<tr>
<td>Command Reference</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>vmstat(1M)</code></td>
<td>When executed in a non-global zone in which the pools facility is enabled, statistics are reported only for the processors in the processor set of the pool to which the zone is bound. Applies to output from the <code>-p</code> option and the <code>page</code>, <code>fault</code>, and <code>cpu</code> report fields.</td>
</tr>
<tr>
<td><code>auditon(2)</code></td>
<td>Added <code>AUDIT_ZONENAME</code> to generate a zone ID token with each audit record.</td>
</tr>
<tr>
<td><code>priocntl(2)</code></td>
<td>Added <code>P_ZONEID id</code> argument.</td>
</tr>
<tr>
<td><code>processor_info(2)</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>p_online(2)</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>pset_bind(2)</code></td>
<td>Added <code>P_ZONEID id</code> type. Added zone to possible choices for <code>P_MYID</code> specification. Added <code>P_ZONEID</code> to valid <code>idtype</code> list in <code>EINVAL</code> error description.</td>
</tr>
<tr>
<td><code>pset_info(2)</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>pset_list(2)</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>pset_setattr(2)</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>sysinfo(2)</code></td>
<td>Changed <code>PRIV_SYS_CONFIG</code> to <code>PRIV_SYS_ADMIN</code>.</td>
</tr>
<tr>
<td><code>umount(2)</code></td>
<td><code>ENOENT</code> is returned if file pointed to by <code>file</code> is not an absolute path.</td>
</tr>
<tr>
<td><code>getloadavg(3C)</code></td>
<td>If the caller is in a non-global zone and the pools facility is enabled, the behavior is equivalent to calling with a <code>psetid</code> of <code>PS_MYID</code>.</td>
</tr>
<tr>
<td><code>getpriority(3C)</code></td>
<td>Added zone IDs to target processes that can be specified. Added zone ID to <code>EINVAL</code> error description.</td>
</tr>
<tr>
<td><code>priv_str_to_set(3C)</code></td>
<td>Added “zone” string for the set of all privileges available within the caller’s zone.</td>
</tr>
<tr>
<td><code>pset_getloadavg(3C)</code></td>
<td>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.</td>
</tr>
<tr>
<td>Command Reference</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>sysconf(3C)</td>
<td>If the caller is in a non-global zone and the pools facility enabled, <code>sysconf(_SC_NPROCESSORS_CONF)</code> and <code>sysconf(_SC_NPROCESSORS_ONLN)</code> return the number of total and online processors in the processor set of the pool to which the zone is bound.</td>
</tr>
<tr>
<td>ucred_get(3C)</td>
<td>Added <code>ucred_getzoneid()</code> function, which returns the zone ID of the process or -1 if the zone ID is not available.</td>
</tr>
<tr>
<td>core(4)</td>
<td>Added <code>n_type: NT_ZONENAME</code>. This entry contains a string that describes the name of the zone in which the process was running.</td>
</tr>
<tr>
<td>pkginfo(4)</td>
<td>Now provides optional parameters and an environment variable in support of zones.</td>
</tr>
<tr>
<td>proc(4)</td>
<td>Added capability to obtain information on processes running in zones.</td>
</tr>
<tr>
<td>audit_syslog(5)</td>
<td>Added <code>in&lt;zone name&gt;</code> field that is used if the zonename audit policy is set.</td>
</tr>
<tr>
<td>privileges(5)</td>
<td>Added <code>PRIV_PROC_ZONE</code>, which allows a process to trace or send signals to processes in other zones. See zones(5).</td>
</tr>
<tr>
<td>if_tcp(7P)</td>
<td>Added zone ioctl() calls.</td>
</tr>
<tr>
<td>cmn_err(9F)</td>
<td>Added zone parameter.</td>
</tr>
<tr>
<td>ddi_cred(9F)</td>
<td>Added <code>crgetzoneid()</code>, which returns the zone ID from the user credential pointed to by <code>cr</code>.</td>
</tr>
</tbody>
</table>
This chapter covers general administration tasks and provides usage examples.

- “Using the ppriv Utility” on page 383
- “Using DTrace in a Non-Global Zone” on page 385
- “Mounting File Systems in Running Non-Global Zones” on page 387
- “Adding Non-Global Zone Access to Specific File Systems in the Global Zone” on page 390
- “Using IP Network Multipathing on a Solaris System With Zones Installed” on page 393
- “Administering Data-Links in Exclusive-IP Non-Global Zones” on page 395
- “Using the Fair Share Scheduler on a Solaris System With Zones Installed” on page 397
- “Using Rights Profiles in Zone Administration” on page 398
- “Backing Up a Solaris System With Installed Zones” on page 399
- “Restoring a Non-Global Zone” on page 402

See Chapter 26, “Solaris Zones Administration (Overview),” for general zone administration topics.

**Using the ppriv Utility**

Use the ppriv utility to display the zone’s privileges.

**How to List Solaris Privileges in the Global Zone**

Use the ppriv utility with the -l option to list the privileges available on the system.

- At the prompt, type `ppriv -l zone` to report the set of privileges available in the zone.

  ```bash
global# ppriv -l zone
  ```
You will see a display similar to this:

```
contract_event
contract_observer
cpc_cpu
.
.
```

▼ **How to List the Non-Global Zone's Privilege Set**

Use the `ppriv` utility with the `-l` option and the expression `zone` to list the zone's privileges.

1. Log into the non-global zone. This example uses a zone named `my-zone`.

2. At the prompt, type `ppriv -l zone` to report the set of privileges available in the zone.

```
my-zone# ppriv -l zone
```

You will see a display similar to this:

```
contract_event
contract_observer
file_chown
.
.
```

▼ **How to List a Non-Global Zone's Privilege Set With Verbose Output**

Use the `ppriv` utility with the `-l` option, the expression `zone`, and the `-v` option to list the zone's privileges.

1. Log into the non-global zone. This example uses a zone named `my-zone`.

2. At the prompt, type `ppriv -l -v zone` to report the set of privileges available in the zone, with a description of each privilege.

```
my-zone# ppriv -lv zone
```
You will see a display similar to this:

```plaintext
contract_event
  Allows a process to request critical events without limitation.
  Allows a process to request reliable delivery of all events on
  any event queue.
contract_observer
  Allows a process to observe contract events generated by
  contracts created and owned by users other than the process’s
  effective user ID.
  Allows a process to open contract event endpoints belonging to
  contracts created and owned by users other than the process’s
  effective user ID.
file_chown
  Allows a process to change a file’s owner user ID.
  Allows a process to change a file’s group ID to one other than
  the process’ effective group ID or one of the process’
  supplemental group IDs.
```

Using DTrace in a Non-Global Zone

Perform the following steps to use DTrace functionality as described in “Running DTrace in a Non-Global Zone” on page 373.

▼ How to Use DTrace

1. Use the zonecfg limitpriv property to add the dtrace_proc and dtrace_user privileges.

   ```plaintext
   global# zonecfg -z my-zone
   zonecfg:my-zone> set limitpriv="default,dtrace_proc,dtrace_user"
   zonecfg:my-zone> exit
   ```

   Note – Depending on your requirements, you can add either privilege, or both privileges.

2. Boot the zone.

   ```plaintext
   global# zoneadm -z my-zone boot
   ```

3. Log in to the zone.

   ```plaintext
   global# zlogin my-zone
   ```
Checking the Status of SMF Services in a Non-Global Zone

To check the status of SMF services in a native non-global zone, use the zlogin command.

▼ How to Check the Status of SMF Services From the Command Line

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 From the command line, type the following to show all services, including disabled ones.
   global# zlogin my-zone svcs -a

See Also For more information, see Chapter 21, “Logging In to Non-Global Zones (Tasks),” and svcs(1).

▼ How to Check the Status of SMF Services From Within a Zone

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Log in to the zone.
   global# zlogin my-zone

3 Run the svcs command with the -a option to show all services, including disabled ones.
   my-zone# svcs -a

See Also For more information, see Chapter 21, “Logging In to Non-Global Zones (Tasks),” and svcs(1).
Mounting File Systems in Running Non-Global Zones

You can mount file systems in a running non-global zone. The following procedures are covered.

- As the global administrator in the global zone, you can import raw and block devices into a non-global zone. After the devices are imported, the zone administrator has access to the disk. The zone administrator can then create a new file system on the disk and perform one of the following actions:
  - Mount the file system manually
  - Place the file system in /etc/vfstab so that it will be mounted on zone boot
- As the global administrator, you can also mount a file system from the global zone into the non-global zone.

▼ SX Only: How to Import Raw and Block Devices by Using zonecfg

This procedure uses the lofi file driver, which exports a file as a block device.

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 Change directories to /usr/tmp.
   global# cd /usr/tmp

3 Create a new UFS file system.
   global# mkfile 10m fsfile

4 Attach the file as a block device.
   The first available slot, which is /dev/lofi/1 if no other lofi devices have been created, is used.
   global# lofiadm -a 'pwd'/fsfile
   You will also get the required character device.

5 Import the devices into the zone my-zone.
   global# zonecfg -z my-zone
   zonecfg:my-zone> add device
   zonecfg:my-zone:device> set match=/dev/rdlofi/1
   zonecfg:my-zone:device> end
   zonecfg:my-zone> add device
   zonecfg:my-zone:device> set match=/dev/lofi/1
   zonecfg:my-zone:device> end
6  Reboot the zone.
   global# zoneadm -z my-zone boot

7  Log in to the zone and verify that the devices were successfully imported.
   my-zone# ls -l /dev/*/lofi/*

   You will see a display that is similar to this:

   brw------- 1 root  sys  147,  1 Jan 7 11:26 /dev/lofi/1
   crw------- 1 root  sys  147,  1 Jan 7 11:26 /dev/rlofi/1

See Also  For more information, see the lofiadm(1M) and lofi(7D) man pages.

▼ How to Mount the File System Manually

   You must be the zone administrator and have the Zone Management profile to perform this
   procedure. This procedure uses the newfs command, which is described in the newfs(1M) man
   page.

1  Become superuser, or have the Zone Management rights profile in your list of profiles.

2  In the zone my-zone, create a new file system on the disk.
   my-zone# newfs /dev/lofi/1

3  Respond yes at the prompt.

   newfs: construct a new file system /dev/rlofi/1: (y/n)? y

   You will see a display that is similar to this:

   /dev/rlofi/1: 20468 sectors in 34 cylinders of 1 tracks, 602 sectors
   10.0MB in 3 cyl groups (16 c/g, 4.70MB/g, 2240 i/g)
   super-block backups (for fsck -F ufs -o b=#) at:
   32, 9664, 19296,

4  Check the file system for errors.
   my-zone# fsck -F ufs /dev/rlofi/1

   You will see a display that is similar to this:

   ** /dev/rlofi/1
   ** Last Mounted on
   ** Phase 1 - Check Blocks and Sizes
   ** Phase 2 - Check Pathnames
   ** Phase 3 - Check Connectivity
Phase 4 - Check Reference Counts
Phase 5 - Check Cyl groups
2 files, 9 used, 9320 free (16 frags, 1163 blocks, 0.2% fragmentation)

5 Mount the file system.
   `my-zone# mount -F ufs /dev/lofi/1 /mnt`

6 Verify the mount.
   `my-zone# grep /mnt /etc/mnttab`
   You will see a display similar to this:
   
   /dev/lofi/1 /mnt ufs
   rw,suid,intr,largefiles,xattr,onerror=panic,zone=foo,dev=24c0001
   1073503869

▼ How to Place a File System in `/etc/vfstab` to Be Mounted When the Zone Boots

This procedure is used to mount the block device `/dev/lofi/1` on the file system path `/mnt`. The block device contains a UFS file system. The following options are used:

- `logging` is used as the mount option.
- `yes` tells the system to automatically mount the file system when the zone boots.
- `/dev/rlofi/1` is the character (or raw) device. The `fsck` command is run on the raw device if required.

1 Become superuser, or have the Zone Management rights profile in your list of profiles.

2 In the zone `my-zone`, add the following line to `/etc/vfstab`:
   `/dev/lofi/1 /dev/rlofi/1 /mnt ufs 2 yes logging`

▼ How to Mount a File System From the Global Zone Into a Non-Global Zone

Assume that a zone has the zonopath `/export/home/my-zone`. You want to mount the disk `/dev/lofi/1` from the global zone into `/mnt` in the non-global zone.

You must be the global administrator in the global zone to perform this procedure.
Adding Non-Global Zone Access to Specific File Systems in the Global Zone

How to Add Access to CD or DVD Media in a Non-Global Zone

This procedure enables you to add read-only access to CD or DVD media in a non-global zone. The Volume Management file system is used in the global zone for mounting the media. A CD or DVD can then be used to install a product in the non-global zone. This procedure uses a DVD named jes_05q4_dvd.

1. Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. To mount the disk into /mnt in the non-global zone, type the following from the global zone:
   
   ```
   global# mount -F ufs /dev/lofi/1 /export/home/my-zone/root/mnt
   ```

   See Also For information about lofi, see the `lofiadm(1M)` and `lofi(7D)` man pages.

   Adding Non-Global Zone Access to Specific File Systems in the Global Zone

   ▼ How to Add Access to CD or DVD Media in a Non-Global Zone

   This procedure enables you to add read-only access to CD or DVD media in a non-global zone. The Volume Management file system is used in the global zone for mounting the media. A CD or DVD can then be used to install a product in the non-global zone. This procedure uses a DVD named jes_05q4_dvd.

   1. Become superuser, or assume the Primary Administrator role.
      To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

   2. Determine whether the Volume Management file system is running in the global zone.
      
      ```
      global# svcs volfs
      STATE STIME FMRI
      online Sep_29 svc:/system/filesystem/volfs:default
      ```

   3. (Optional) If the Volume Management file system is not running in the global zone, start it.
      
      ```
      global# svcadm volfs enable
      ```

   4. Insert the media.

   5. Check for media in the drive.
      
      ```
      global# volcheck
      ```

   6. Test whether the DVD is automounted.
      
      ```
      global# ls /cdrom
      ```
Adding Non-Global Zone Access to Specific File Systems in the Global Zone

You will see a display similar to the following:

```
cdrom  cdrom1  jes_05q4_dvd
```

7 Loopback mount the file system with the options `ro,nodevices` (read-only and no devices) in the non-global zone.

```
global# zonecfg -z my-zone
zonecfg:my-zone> add fs
zonecfg:my-zone:fs> set dir=/cdrom
zonecfg:my-zone:fs> set special=/cdrom
zonecfg:my-zone:fs> set type=lofs
zonecfg:my-zone:fs> add options [ro,nodevices]
zonecfg:my-zone:fs> end
zonecfg:my-zone> commit
zonecfg:my-zone> exit
```

8 Reboot the non-global zone.

```
global# zoneadm -z my-zone reboot
```

9 Use the `zoneadm list` command with the `-v` option to verify the status.

```
global# zoneadm list -v
```

You will see a display that is similar to the following:

```
ID   NAME  STATUS  PATH          BRAND  IP
0    global running /          native  shared
1    my-zone running /export/home/my-zone native  shared
```

10 Log in to the non-global zone.

```
global# zlogin my-zone
```

11 Verify the DVD-ROM mount.

```
my-zone# ls /cdrom
```

You will see a display similar to this:

```
cdrom  cdrom1  jes_05q4_dvd
```

12 Install the product as described in the product installation guide.

13 Exit the non-global zone.

```
my-zone# exit
```
Tip – You might want to retain the /cdrom file system in your non-global zone. The mount will always reflect the current contents of the CD-ROM drive, or an empty directory if the drive is empty.

14 (Optional) If you want to remove the /cdrom file system from the non-global zone, use the following procedure.

```
global# zonecfg -z my-zone
zonecfg:my-zone> remove fs dir=/cdrom
zonecfg:my-zone> commit
zonecfg:my-zone> exit
```

▼ How to Add a Writable Directory under /usr in a Non-Global Zone

In a native sparse root zone, /usr is mounted read-only from the global zone. You can use this procedure to add a writable directory, such as /usr/local, under /usr in your zone.

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Create the directory /usr/local in the global zone.
   global# mkdir -p /usr/local

3 Specify a directory in the global zone to serve as the backing store for the zone’s /usr/local directory.
   global# mkdir -p /storage/local/my-zone

4 Edit the configuration for the zone my-zone.
   global# zonecfg -z my-zone

5 Add the loopback-mounted filesystem.
   zonecfg:my-zone> add fs
   zonecfg:my-zone:fs> set dir=/usr/local
   zonecfg:my-zone:fs> set special=/storage/local/my-zone
   zonecfg:my-zone:fs> set type=lofs
   zonecfg:my-zone:fs> end
   zonecfg:my-zone:fs> commit
   zonecfg:my-zone:fs> exit
6 Boot the zone.

How to Export Home Directories in the Global Zone Into a Non-Global Zone

This procedure is used to export home directories or other file systems from the global zone into non-global zones on the same system.

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Add the loopback-mounted filesystem.
   global# zonecfg -z my-zone
   zonecfg:my-zone> add fs
   zonecfg:my-zone:fs> set dir=/export/home
   zonecfg:my-zone:fs> set special=/export/home
   zonecfg:my-zone:fs> set type=lofs
   zonecfg:my-zone:fs> set options=nodevices
   zonecfg:my-zone:fs> end
   zonecfg:my-zone> commit
   zonecfg:my-zone> exit

3 Add the following line to the zone’s /etc/auto_home file:
   $HOST:/export/home&

Using IP Network Multipathing on a Solaris System With Zones Installed

How to Use IP Network Multipathing in Exclusive-IP Non-Global Zones

IP Network Multipathing (IPMP) in an exclusive-IP zone is configured as it is in the global zone.

You can configure one or more physical interfaces into an IP multipathing group, or IPMP group. After configuring IPMP, the system automatically monitors the interfaces in the IPMP group for failure. If an interface in the group fails or is removed for maintenance, IPMP
automatically migrates, or fails over, the failed interface's IP addresses. The recipient of these addresses is a functioning interface in the failed interface's IPMP group. The failover feature of IPMP preserves connectivity and prevents disruption of any existing connections. Additionally, IPMP improves overall network performance by automatically spreading out network traffic across the set of interfaces in the IPMP group. This process is called load spreading.

1  **Become superuser, or assume the Primary Administrator role.**

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2  **Configure IPMP groups as described in “Configuring IPMP Groups” in System Administration Guide: Network Interfaces and Network Virtualization.**

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

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

Each address, or logical interface, should be associated with a non-global zone when you configure the zone. See “Using the `zonecfg` Command” on page 225 and “How to Configure the Zone” on page 249 for instructions.

This procedure accomplishes the following:

- The cards bge0 and hme0 are configured together in a group.
- Address 192.168.0.1 is associated with the non-global zone *my-zone*.
- The bge0 card is set as the physical interface. Thus, the IP address is hosted in the group that contains the bge0 and hme0 cards.

In a running zone, you can use the `ifconfig` command to make the association. See “Shared-IP Network Interfaces” on page 361 and the `ifconfig(1M)` man page.

You must be the global administrator in the global zone to perform this procedure.

1  **Become superuser, or assume the Primary Administrator role.**

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2  **In the global zone, configure IPMP groups as described in “Configuring IPMP Groups” in System Administration Guide: Network Interfaces and Network Virtualization.**
Use the `zonecfg` command to configure the zone. When you configure the `net` resource, add address 192.168.0.1 and physical interface `bge0` to the zone `my-zone`:

```
zonecfg:my-zone> add net
zonecfg:my-zone:net> set address=192.168.0.1
zonecfg:my-zone:net> set physical=bge0
zonecfg:my-zone:net> end
```

Only `bge0` would be visible in non-global zone `my-zone`.

**More Information** If `bge0` Subsequently Fails

If `bge0` subsequently fails and the `bge0` data addresses fail over to `hme0` in the global zone, the `my-zone` addresses migrate as well.

If address 192.168.0.1 moves to `hme0`, then only `hme0` would now be visible in non-global zone `my-zone`. This card would be associated with address 192.168.0.1, and `bge0` would no longer be visible.

### Administering Data-Links in Exclusive-IP Non-Global Zones

The `dladm` command is used from the global zone to administer data-links.

**How to Use `dladm show-linkprop`**

The `dladm` command can be used with the `show-linkprop` subcommand to show the assignment of data-links to running exclusive-IP zones.

You must be the global administrator in the global zone to administer data-links.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Show the assignment of data-links on the system.**

   ```
   global# dladm show-linkprop
   ```

**Example 27–1 Using `dladm` With the `show-linkprop` subcommand**

1. In the first screen, zone 49bge, which is assigned `bge0` has not been booted

   ```
   global# dladm show-linkprop
   ```

<table>
<thead>
<tr>
<th>LINK</th>
<th>PROPERTY</th>
<th>VALUE</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>bge0</td>
<td>zone</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
How to Use \texttt{dladm set-linkprop}

The \texttt{dladm} command can be used with the \texttt{set-linkprop} subcommand to temporarily assign data-links to running exclusive-IP zones. Persistent assignment must be made through the \texttt{zonecfg} command.

You must be the global administrator in the global zone to administer data-links.

1. \textbf{Become superuser, or assume the Primary Administrator role.}

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. \textbf{Use} \texttt{dladm set-linkprop with the \texttt{-t} to add bge0 to a running zone called excl.}

\begin{verbatim}
global# dladm set-linkprop -t -p zone=excl bge0
\end{verbatim}

\textbf{Tip} – The \texttt{-p} option produces a display using a stable machine-parseable format.
How to Use `dladm reset-linkprop`

The `dladm` command can be used with the `reset-linkprop` subcommand to reset the bge0 link value to unassigned.

1 **Become superuser, or assume the Primary Administrator role.**

   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 **Use `dladm reset-linkprop` with the `-t` to undo the zone assignment of the bge0 device.**

   ```
   global# dladm set-linkprop -t -p zone=excl bge0
   LINK PROPERTY  VALUE  DEFAULT  POSSIBLE
   bge0  zone     excl     --       --
   ```

   **Tip** – The `-p` option produces a display using a stable machine-parseable format.

**Troubleshooting**

If the running zone is using the device, the reassignment fails and an error message is displayed. See "Exclusive-IP Zone Is Using Device, so `dladm reset-linkprop` Fails" on page 405.

Using the Fair Share Scheduler on a Solaris System With Zones Installed

Limits specified through the `prctl` command are not persistent. The limits are only in effect until the system is rebooted. To set shares in a zone permanently, see “How to Configure the Zone” on page 249 and “How to Set `zone.cpu-shares` in the Global Zone” on page 260.

How to Set FSS Shares in the Global Zone Using the `prctl` Command

The global zone is given one share by default. You can use this procedure to change the default allocation. Note that you must reset shares allocated through the `prctl` command whenever you reboot the system.

You must be the global administrator in the global zone to perform this procedure.

1 **Become superuser, or assume the Primary Administrator role.**

   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.
Use the `prctl` utility to assign two shares to the global zone:

```bash
# prctl -n zone.cpu-shares -v 2 -r -i zone global
```

(Optional) To verify the number of shares assigned to the global zone, type:

```bash
# prctl -n zone.cpu-shares -i zone global
```

For more information on the `prctl` utility, see the `prctl(1)` man page.

### How to Change the `zone.cpu-shares` Value in a Zone Dynamically

This procedure can be used for any zone, not just the global zone.

1. Become superuser, or assume the Primary Administrator role.
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. Use the `prctl` command to specify a new value for `cpu-shares`.

```bash
# prctl -n zone.cpu-shares -r -v value -i zone zonename
```

`idtype` is either the `zonename` or the `zoneid`. `value` is the new value.

### Using Rights Profiles in Zone Administration

This section covers tasks associated with using rights profiles in non-global zones.

#### How to Assign the Zone Management Profile

The Zone Management profile grants the power to manage all of the non-global zones on the system to a user.

You must be the global administrator in the global zone to perform this procedure.

1. Become superuser, or assume the Primary Administrator role.
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. Create a role that includes the Zone Management rights profile, and assign the role to a user.
   
   To create and assign the role by using the Solaris Management Console, see “Configuring RBAC (Task Map)” in *System Administration Guide: Security Services*. Refer to the task “How to Create and Assign a Role By Using the GUI.”
To create and assign the role on the command line, see "Managing RBAC" in System Administration Guide: Security Services. Refer to the task "How to Create a Role From the Command Line."

**Example—Using Profile Shells With Zone Commands**

You can execute zone commands in a profile using the `pfexec` program. The program executes commands with the attributes specified by the user's profiles in the `exec_attr` database. The program is invoked by the profile shells `pfksh`, `pfcsh`, and `pfsh`.

Use the `pfexec` program to log in to a zone, for example, `my-zone`.

```
machine$ pfexec zlogin my-zone
```

**Backing Up a Solaris System With Installed Zones**

The following procedures can be used to back up files in zones. Remember to also back up the zones' configuration files.

**How to Use `ufsdump` to Perform Backups**

You can perform full or incremental backups using the `ufsdump` command. This procedure backs up the zone `/export/my-zone` to `/backup/my-zone`. `ufsdump`, where `my-zone` is replaced with the name of a zone on your system. You might want to have a separate file system, for example, a file system mounted on `/backup`, to hold the backups.

1. **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2. **(Optional) Shut down the zone to put the zone in a quiescent state and to avoid creating backups of shared file systems.**
   ```
global# zlogin -S my-zone init 0
   ```

3. **Check the zone's status.**
   ```
global# zoneadm list -cv
   ```

   You will see a display similar to the following:

```plaintext
<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>my-zone</td>
<td>installed</td>
<td>/export/home/my-zone</td>
<td>native</td>
<td>shared</td>
</tr>
</tbody>
</table>
```
4 Perform the backup.

    global# ufsdump 0f /backup/my-zone.ufsdump /export/my-zone

You will see a display similar to the following:

DUMP: Date of this level 0 dump: Wed Aug 10 16:13:52 2005
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdsk/c0t0d0s0 (bird:/) to /backup/my-zone.ufsdump.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Writing 63 Kilobyte records
DUMP: Estimated 363468 blocks (174.47MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 369934 blocks (180.63MB) on 1 volume at 432 KB/sec
DUMP: DUMP IS DONE

5 Boot the zone.

    global# zoneadm -z my-zone boot

How to Create a UFS Snapshot Using fssnap

This approach uses the fssnap command, which creates a temporary image of a file system intended for backup operations.

This method can be used to provide a clean, consistent backup of the zone files only, and it can be executed while zones are running. However, it is a good idea to suspend or checkpoint active applications that are updating files when the snapshot is created. An application updating files when the snapshot is created might leave these files in an internally inconsistent, truncated, or otherwise unusable state.

In the example procedure below, note the following:

- There is a zone named my-zone under /export/home.
- /export/home is a separate file system.

Before You Begin The destination backup is /backup/my-zone.ufs. You must create the directory backup under /.

1 Become superuser, or assume the Primary Administrator role.

   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Create the snapshot.

    global# fssnap -o bs=/export /export/home
You will see a display similar to the following:

```
dev/fssnap/0
```

3 Mount the snapshot.
```
global# mount -o ro /dev/fssnap/0 /mnt
```

4 Back up my-zone from the snapshot.
```
global# ufsdump 0f /backup/my-zone.ufsdump /mnt/my-zone
```

You will see a display similar to the following:

```
DUMP: Date of this level 0 dump: Thu Oct 06 15:13:07 2005
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rfssnap/0 (pc2:/mnt) to /backup/my-zone.ufsdump.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Writing 32 Kilobyte records
DUMP: Estimated 176028 blocks (85.95MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 175614 blocks (85.75MB) on 1 volume at 2731 KB/sec
DUMP: DUMP IS DONE
```

5 Unmount the snapshot.
```
global# umount /mnt
```

6 Delete the snapshot.
```
global# fssnap -d /dev/fssnap/0
```

Note that the snapshot is also removed from the system when the system is rebooted.

How to Use find and cpio to Perform Backups

1 Become superuser, or assume the Primary Administrator role.
To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 Change directories to the root directory.
```
global# cd /
```

3 Back up my-zone files that are not loopback mounted to /backup/my-zone.cpio.
```
global# find export/my-zone -fstype lofs -prune -o -local | cpio -oc -O /backup/my-zone.cpio type as one line
```
4 Verify the results.
   
   global# ls -l backup/my-zone.cpio
   
   You will see a display similar to the following:
   
   -rwxr-xr-x 1 root root 99680256 Aug 10 16:13 backup/my-zone.cpio

▼ How to Print a Copy of a Zone Configuration

You should create backup files of your non-global zone configurations. You can use the backups to recreate the zones later if necessary. Create the copy of the zone’s configuration after you have logged in to the zone for the first time and have responded to the sysidtool questions. This procedure uses a zone named my-zone and a backup file named my-zone.config to illustrate the process.

1 Become superuser, or assume the Primary Administrator role.
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Print the configuration for the zone my-zone to a file named my-zone.config.
   
   global# zonecfg -z my-zone export > my-zone.config

Restoring a Non-Global Zone

▼ How to Restore an Individual Non-Global Zone

You can use the backup files of your non-global zone configurations to restore non-global zones, if necessary. This procedure uses a zone named my-zone and a backup file named my-zone.config to illustrate the process of restoring a zone.

1 Become superuser, or assume the Primary Administrator role.
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Specify that my-zone.config be used as the zonecfg command file to recreate the zone my-zone.
   
   global# zonecfg -z my-zone -f my-zone.config

3 Install the zone.
   
   global# zoneadm -z my-zone install
To prevent the system from displaying the `sysidtool` questions upon initial zone login, delete the file `/zonepath/root/etc/.UNCONFIGURED`, for example:

```sh
global# rm /export/home/my-zone/root/etc/.UNCONFIGURED
```

If you have any zone-specific files to restore, such as application data, manually restore (and possibly hand-merge) files from a backup into the newly created zone’s root file system.
This chapter contains zones troubleshooting information.

**Information on Zones in the OpenSolaris 2009.06 Release**


**Exclusive-IP Zone Is Using Device, so `dladm reset-linkprop` Fails**

If the following error message is displayed:

```
    dladm: warning: cannot reset link property 'zone' on 'bge0': operation failed
```

Referring to "How to Use `dladm reset-linkprop` on page 397, the attempt to use `dladm reset-linkprop` failed. The running zone `excl` is using the device, which was assigned by executing `ifconfig bge0 plumb` inside the zone.

To reset the value, use the procedure `ifconfig bge0 unplumb` inside the zone and rerun the `dladm` command.

**Incorrect Privilege Set Specified in Zone Configuration**

If the zone’s privilege set contains a disallowed privilege, is missing a required privilege, or includes an unknown privilege name, an attempt to verify, ready, or boot the zone will fail with an error message such as the following:

```
    zonecfg:zone5> set limitpriv="basic"
    .
```


Zone Administrator Mounting Over File Systems Populated by the Global Zone

The presence of files within a file system hierarchy when a non-global zone is first booted indicates that the file system data is managed by the global zone. When the non-global zone was installed, a number of the packaging files in the global zone were duplicated inside the zone. These files must reside under the zonepath directly. If the files reside under a file system created by a zone administrator on disk devices or ZFS datasets added to the zone, packaging and patching problems could occur.

The issue with storing any of the file system data that is managed by the global zone in a zone-local file system can be described by using ZFS as an example. If a ZFS dataset has been delegated to a non-global zone, the zone administrator should not use that dataset to store any of the file system data that is managed by the global zone. The configuration could not be patched or upgraded correctly.

For example, a ZFS delegated dataset should not be used as a /var file system. The Solaris operating system delivers core packages that install components into /var. These packages have to access /var when they are upgraded or patched, which is not possible if /var is mounted on a delegated ZFS dataset.

File system mounts under parts of the hierarchy controlled by the global zone are supported. For example, if an empty /usr/local directory exists in the global zone, the zone administrator can mount other contents under that directory.

You can use a delegated ZFS dataset for file systems that do not need to be accessed during patching or upgrade, such as /export in the non-global zone.

netmasks Warning Displayed When Booting Zone

If you see the following message when you boot the zone as described in “How to Boot a Zone” on page 278:

```bash
# zoneadm -z my-zone boot
zoneadm: zone 'my-zone': WARNING: hme0:1: no matching subnet found in netmasks(4) for 192.168.0.1; using default of 255.255.255.0.
```
The message is only a warning, and the command has succeeded. The message indicates that the system was unable to find the netmask to be used for the IP address specified in the zone’s configuration.

To stop the warning from displaying on subsequent reboots, ensure that the correct netmasks databases are listed in the /etc/nsswitch.conf file in the global zone and that at least one of these databases contains the subnet and netmasks to be used for the zone my-zone.

For example, if the /etc/inet/netmasks file and the local NIS database are used for resolving netmasks in the global zone, the appropriate entry in /etc/nsswitch.conf is as follows:

netmasks: files nis

The subnet and corresponding netmask information for the zone my-zone can then be added to /etc/inet/netmasks for subsequent use.

For more information about the netmasks command, see the netmasks(4) man page.

Zone Does Not Halt

In the event that the system state associated with the zone cannot be destroyed, the halt operation will fail halfway. This leaves the zone in an intermediate state, somewhere between running and installed. In this state there are no active user processes or kernel threads, and none can be created. When the halt operation fails, you must manually intervene to complete the process.

The most common cause of a failure is the inability of the system to unmount all file systems. Unlike a traditional Solaris system shutdown, which destroys the system state, zones must ensure that no mounts performed while booting the zone or during zone operation remain once the zone has been halted. Even though zoneadm makes sure that there are no processes executing in the zone, the unmount operation can fail if processes in the global zone have open files in the zone. Use the tools described in the proc(1) (see pfiles) and fuser(1M) man pages to find these processes and take appropriate action. After these processes have been dealt with, reinvoking zoneadm halt will completely halt the zone.

Resolving Problems With a zoneadm attach Operation

▼ Patches and Packages Are Out of Sync

The target system must be running the same or later versions of the following required operating system packages and patches as those installed on the original host.

- Packages that deliver files under an inherit-pkg-dir resource
Packages where SUNW_PKG_ALLZONES=true

1. If packages and patches are different between the original host and the new host, you might see a display similar to the following:

   host2# zoneadm -z my-zone attach
   These packages installed on the source system are inconsistent with this system:  
   SUNWgnome-libs (2.6.0,REV=101.0.3.2005.12.06.20.27) version mismatch  
   (2.6.0,REV=101.0.3.2005.12.19.21.22)  
   SUNWudaplr (11.11,REV=2005.12.13.01.06) version mismatch  
   (11.11,REV=2006.01.03.00.45)  
   SUNWradpu320 (11.10.0,REV=2005.01.21.16.34) is not installed  
   SUNWaudf (11.11,REV=2005.12.13.01.06) version mismatch  
   (11.11,REV=2006.01.03.00.45)  
   NCRos86r (11.10.0,REV=2005.01.17.23.31) is not installed  
   These packages installed on this system were not installed on the source system:  
   SUNWukspfw (11.11,REV=2006.01.03.00.45) was not installed  
   SUNWsmcmd (1.0,REV=2005.12.14.01.53) was not installed  
   These patches installed on the source system are inconsistent with this system:  
   120081 is not installed  
   118844 is not installed  
   118344 is not installed  
   These patches installed on this system were not installed on the source system:  
   118669 was not installed  
   118668 was not installed  
   116299 was not installed

2. To migrate the zone successfully, use one of the following methods:

   - Update the new host with the correct packages and patches so that this content is the same on both systems. For more information, see Chapter 24, “About Packages and Patches on a Solaris System With Zones Installed (Overview),” and Chapter 25, “Adding and Removing Packages and Patches on a Solaris System With Zones Installed (Tasks)”

   - If the new host has later versions of the zone-dependent packages or their associated patches, use zoneadm attach with the -u option to update those packages within the zone to match the new host. See “About Migrating a Zone” on page 306.

▼ Operating System Releases Do Not Match

To migrate the zone successfully, install the same Solaris release that is running on the original host on a system with the same architecture.

1. Verify the Solaris release running on the original system.

   host1# uname -a
2 Install the same release on the new host.
   Refer to the Solaris installation documentation on docs.sun.com.

▼ Machine Architectures Do Not Match
   To migrate the zone successfully, use the -u option to zoneadm attach.

1 Verify the system architecture on both systems.
   host1# uname -a

2 If the architectures are different, use the -u option to zoneadm attach to perform the attach.
   host2# zoneadm -z my-zone attach -u
   For more information, see “How to Migrate A Non-Global Zone” on page 307.
Linux Branded Zones

BrandZ provides the framework to create non-global branded zones that contain non-native operating environments. Branded zones are used on the Solaris Operating System to run applications. The first brand available was the lx brand, Solaris Containers for Linux Applications. The lx brand provides a Linux environment for your applications and runs on x86 and x64 machines.
About Branded Zones and the Linux Branded Zone

The branded zones facility in the Solaris™ Operating System is a simple extension of Solaris Zones. This chapter discusses the branded zones concept and the lx brand, which implements Linux branded zones functionality. Linux branded zones are also known as Solaris Containers for Linux Applications.

**Note** – Although you can configure and install lx brand zones on a Solaris Trusted Extensions system that has labels enabled, you cannot boot branded zones on this system configuration. Only the labeled brand can be booted on an OpenSolaris system configuration.

## About Using Zones on a Solaris System

See Chapter 15, “Introduction to Solaris Zones,” for general information on the use of zones on a Solaris system.

You should be familiar with the following zones and resource management concepts:

- The global zone and the non-global zone, described in “How Zones Work” on page 204.
- The global administrator and the zone administrator, described in “How Non-Global Zones Are Administered” on page 206 and “How Non-Global Zones Are Created” on page 206.
- The zone state model, discussed in “Non-Global Zone State Model” on page 207.
- The zone isolation characteristics covered in “Non-Global Zone Characteristics” on page 209.
- Privileges, described in “Privileges in a Non-Global Zone” on page 367.
- Networking, described in “Networking in Shared-IP Non-Global Zones” on page 360.
- The Solaris Container concept, which is the use of resource management features, such as resource pools, with zones. The use and interaction of zones and resource management features are described in “Using Resource Management Features With Non-Global Zones” on page 210, “Setting Zone-Wide Resource Controls” on page 222, Chapter 26, “Solaris
The fair share scheduler (FSS), a scheduling class that enables you to allocate CPU time based on shares, is covered in Chapter 8, "Fair Share Scheduler (Overview)," and Chapter 9, "Administering the Fair Share Scheduler (Tasks)."

The resource capping daemon (rcapd), which can be used from the global zone to control resident set size (RSS) usage of branded zones. The property of the zonecfg capped-memory resource sets the max-rss for a zone. This value is enforced by rcapd running in the global zone. For more information, see Chapter 10, "Physical Memory Control Using the Resource Capping Daemon (Overview)," Chapter 11, "Administering the Resource Capping Daemon (Tasks)," and the rcapd(1M) man page.

The Glossary provides definitions for terms used with zones and resource management features.

Any additional information required to use branded zones on your system is provided in this part of the guide.

Note – The following chapters in this guide are not applicable to branded zones:

- Chapter 24, “About Packages and Patches on a Solaris System With Zones Installed (Overview)"
- Chapter 25, "Adding and Removing Packages and Patches on a Solaris System With Zones Installed (Tasks)"

Branded Zones Technology

The Solaris Zones infrastructure is documented in this manual in Part II. By default, a non-global zone has the same characteristics as operating system in the global zone, which is running the Solaris 10 Operating System or later Solaris 10 release. These native non-global zones and the global zone share their conformance to standards, runtime behavior, command sets, and performance traits in common. The branded zone (BrandZ) framework extends the zones infrastructure to include the creation of brands, or alternative sets of runtime behaviors. The term brand can refer to a wide range of operating environments. For example, the non-global zone can emulate another version of the Solaris Operating System, or an operating environment such as Linux. Or, it might augment the native brand behaviors with additional characteristics or features. Every zone is configured with an associated brand.

A brand can provide a simple or a complex environment. For example, a simple environment could replace the standard Solaris utilities with their GNU equivalents. A complex environment could provide a complete Linux user space which supports the execution of Linux applications.
The brand defines the operating environment that can be installed in the zone and determines how the system will behave within the zone so that the non-native software installed in the zone functions correctly. In addition, a zone’s brand is used to identify the correct application type at application launch time. All branded zone management is performed through extensions to the native zones structure. Most administration procedures are identical for all zones.

You can change the brand of a zone in the configured state. Once a branded zone has been installed, the brand cannot be changed or removed.

BrandZ extends the zones tools in the following ways:

- The `zonecfg` command is used to set a zone’s brand type when the zone is configured.
- The `zoneadm` command is used to report a zone’s brand type as well as administer the zone.

**Note** – You can change the brand of a zone in the configured state. Once a branded zone has been installed, that brand cannot be changed or removed.

**Processes Running in a Branded Zone**

Branded zones provide a set of interposition points in the kernel that are only applied to processes executing in a branded zone.

- These points are found in such paths as the `syscall` path, the process loading path, and the thread creation path.
- At each of these points, a brand can choose to supplement or replace the standard Solaris behavior.

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

**Branded Zone Device Support**

The devices supported by each zone are documented in the man pages and other documentation for that brand. Device support is defined by the brand. A brand can choose to disallow the addition of any unsupported or unrecognized devices.

**Branded Zone File System Support**

The file systems required for a branded zone are defined by the brand.
Privileges in a Branded Zone

The privileges available in a branded zone are defined by the brand. For more information about privileges, see “Privileges in a Non-Global Zone” on page 367 and “Configurable Privileges in an \( \text{lx} \) Branded Zone” on page 428.

About the \( \text{lx} \) Brand

The \( \text{lx} \) brand uses the branded zones framework to enable Linux binary applications to run unmodified on a machine with a Solaris Operating System kernel.

The machine must have one of the following supported i686 processor types:

- Intel
  - Pentium Pro
  - Pentium II
  - Pentium III
  - Celeron
  - Xeon
  - Pentium 4
  - Pentium M
  - Pentium D
  - Pentium Extreme Edition
  - Core
  - Core 2

- AMD
  - Opteron
  - Athlon XP
  - Athlon 64
  - Athlon 64 X2
  - Athlon FX
  - Duron
  - Sempron
  - Turion 64
  - Turion 64 X2

Supported Linux Distributions

The \( \text{lx} \) brand includes the tools necessary to install a CentOS 3.x or Red Hat Enterprise Linux 3.x distribution inside a non-global zone. Versions 3.5 to 3.8 of each distribution are supported. The brand supports the execution of 32-bit Linux applications on x86 and x64 machines running the Solaris system in either 32-bit or 64-bit mode.
The `lx` brand emulates the system call interfaces provided by the Linux 2.4.21 kernel, as modified by Red Hat in the RHEL 3.x distributions. This kernel provides the system call interfaces consumed by the glibc version 2.3.2 released by Red Hat.

In addition, the `lx` brand partially emulates the Linux `/dev` and `/proc` interfaces.

**Caution** – Note that you must maintain a supported configuration if you add packages to an `lx` branded zone. See “About Maintaining a Supported Configuration” on page 485 for more information.

### Application Support

The Solaris system imposes no limit on the number of Linux applications you can run in an `lx` branded zone. Sufficient memory must be available. Also see “System and Space Requirements” on page 421.

Regardless of the underlying kernel, only 32-bit Linux applications are able to run.

The `lx` zone supports only user-level Linux applications. You cannot use Linux device drivers, Linux kernel modules, or Linux file systems from inside an `lx` zone.

See [http://opensolaris.org/os/community/brandz/applications](http://opensolaris.org/os/community/brandz/applications) for a list of some applications that have been successfully run under the `lx` brand. See "How to Install an Application in an `lx` Branded Zone" on page 486 for an example of installing an application.

You cannot run Solaris applications inside an `lx` zone. However, the `lx` zone enables you to use the Solaris system to develop, test, and deploy Linux applications. For example, you can place a Linux application in an `lx` zone and analyze it using Solaris tools run from the global zone. You can then make improvements and deploy the tuned application on a native Linux system.

### Debugging Tools

Solaris debugging tools such as DTrace and mdb can be applied to Linux processes executing inside the zone, but the tools themselves must be running in the global zone. Any core files generated are produced in the Solaris format and can only be debugged with Solaris tools.

DTrace is enabled for Linux applications by the DTrace `lxsyscall` dynamic tracing provider. The provider acts like the DTrace `syscall` provider. The `lxsyscall` provider provides probes that fire whenever a thread enters or returns from a Linux system call entry point.

For more information on debugging options, see the Solaris Dynamic Tracing Guide, and the `dtrace(1M)` and `mdb(1)` man pages. The Solaris Dynamic Tracing Guide describes the public documented interfaces available for the DTrace facility. The documentation about the `syscall` provider can be used for the `lxsyscall` provider.
Note – Because NFS is dependent on name services, which are zone specific, you cannot access any NFS file system that is mounted outside of the current zone. Thus, you cannot debug NFS-based Linux processes from the global zone.

Commands and Other Interfaces

The commands identified in the following table provide the primary administrative interface to the zones facility.

TABLE 29-1 Commands and Other Interfaces Used With \x Branded Zones

<table>
<thead>
<tr>
<th>Command Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>zlogin(1)</td>
<td>Log in to a non-global zone</td>
</tr>
<tr>
<td>zoneadm(1M)</td>
<td>Administers zones on a system</td>
</tr>
<tr>
<td>zonecfg(1M)</td>
<td>Used to set up a zone configuration</td>
</tr>
<tr>
<td>getzoneid(3C)</td>
<td>Used to map between zone ID and name</td>
</tr>
<tr>
<td>brands(5)</td>
<td>Provides description of branded zones facility</td>
</tr>
<tr>
<td>\x(5)</td>
<td>Provides description of Linux branded zones</td>
</tr>
<tr>
<td>zones(5)</td>
<td>Provides description of zones facility</td>
</tr>
<tr>
<td>\x_systrace(7D)</td>
<td>DTrace Linux system call tracing provider</td>
</tr>
<tr>
<td>zcons(7D)</td>
<td>Zone console device driver</td>
</tr>
</tbody>
</table>

The zoneadm daemon is the primary process for managing the zone’s virtual platform. The man page for the zoneadm daemon is zoneadm(1M). The daemon does not constitute a programming interface.

Note – Table 26–5 covers commands that can be used in the global zone to display information about all non-global zones, including branded zones. Table 26–4 covers commands used with the resource capping daemon.
### Setting Up \(lx\) Branded Zones on Your System (Task Map)

The following table provides an overview of the tasks that are involved in setting up \(lx\) zones on your system for the first time.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify each 32-bit Linux application that you would like to run in a zone.</td>
<td>Assess the system needs of the application.</td>
<td>Refer to your business goals and to your system documentation if necessary.</td>
</tr>
</tbody>
</table>
| Determine how many zones to configure. | Assess:  
- The number of Linux applications you intend to run.  
- The disk space requirements for Linux branded zones.  
| Determine whether you will use resource pools with your zone to create a container. | If you are using resource pools, configure the pools before you configure zones.  
Note that you can add zone-wide resource controls and pool functionality to a zone quickly by using zonecfg properties. | See "How to Configure the \(lx\) Branded Zone" on page 440, Chapter 13, "Creating and Administering Resource Pools (Tasks)." |
| Perform the preconfiguration tasks. | Determine the zone name and the zone path for each zone. If network connectivity is required, obtain IP addresses. Determine the scheduling class for the zone. Determine the set of privileges that processes inside the zone should be limited to, if the standard default set is not sufficient. | For information on the zone name, zone path, IP addresses, and scheduling class, see "\(lx\) Branded Zone Configuration Components" on page 422. For a listing of default privileges and privileges that can be configured in a non-global zone, see "Privileges in a Non-Global Zone" on page 367.  
For information on resource pool association, see "How Zones Work" on page 204 and "How to Configure the \(lx\) Branded Zone" on page 440. |
| Develop configurations. | Configure non-global zones. | See "Configuring, Verifying, and Committing a Zone" on page 249 and the zonecfg(1M) man page. |
As global administrator, verify and install configured zones. Zones must be verified and installed prior to booting the zone. You must obtain a Linux distribution before you install a Linux branded zone.

See Chapter 32, “About Installing, Booting, Halting, Cloning, and Uninstalling 1x Branded Zones (Overview),” and Chapter 33, “Installing, Booting, Halting, Uninstalling and Cloning 1x Branded Zones (Tasks).”

As global administrator, boot the non-global zones. Boot each zone to place the zone in the running state.

See Chapter 33, “Installing, Booting, Halting, Uninstalling and Cloning 1x Branded Zones (Tasks).”

Prepare the new zone for production use. Create user accounts, add additional software, and customize the zone’s configuration using standard Linux system administration tools and methodologies from within the zone.

Refer to the documentation you use to set up a newly installed machine and install applications. Special considerations applicable to a system with zones installed are covered in this guide.
Planning the \( \text{x} \) Branded Zone Configuration
(Overview)

This chapter describes what you need to do before you can configure an \( \text{x} \) branded zone on your x64 or x86 based system. This chapter also describes how to use the zonecfg command.

System and Space Requirements

The following primary machine considerations are associated with the use of \( \text{x} \) branded zones.

- The machine must be either x64 or x86 based.
- Sufficient disk space to hold the files that are unique within each \( \text{x} \) zone must be available. The disk space requirements for an \( \text{x} \) zone are determined by the size and number of RPMs, or Linux packages, that are installed.
- The \( \text{x} \) brand supports only the whole root model, so each installed zone will have its own copy of every file.

There are no limits on how much disk space can be consumed by a zone. The global administrator is responsible for space restriction. The global administrator must ensure that local storage is sufficient to hold a non-global zone’s root file system. Given sufficient storage, even a small uniprocessor system can support a number of zones running simultaneously.

Restricting the Size of the Branded Zone

The following options can be used to restrict zone size:

- You can place the zone on a lofi-mounted partition. This action will limit the amount of space consumed by the zone to that of the file used by lofi. For more information, see the lofiadm(1M) and lofi(7D) man pages.
- You can use soft partitions to divide disk slices or logical volumes into partitions. You can use these partitions as zone roots, and thus limit per-zone disk consumption. The soft partition limit is 8192 partitions. For more information, see Chapter 12, “Soft Partitions (Overview),” in Solaris Volume Manager Administration Guide.
You can use the standard partitions of a disk for zone roots, and thus limit per-zone disk consumption.

### Branded Zone Network Address

Each zone that requires network connectivity has one or more unique IP addresses. IPv4 addresses are supported. You must assign an IPv4 address for the zone. For more information, see "Branded Zone Network Address" on page 422.

### Branded Zone Configuration Process

The `zonecfg` command is used to:

- Set the brand for the zone
- Create the configuration for the `lx` zone
- Verify the configuration to determine whether the specified resources and properties are legal and internally consistent on a hypothetical x86 or x64 based system
- Perform a brand-specific verification. The verification ensures the following:
  - The zone cannot have any inherited package directories, ZFS datasets, or added devices.
  - If the zone is configured to use audio, the specified devices (if any) must be `none`, `default`, or a single digit.

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

- Ensures that a zone path is specified
- Ensures that all of the required properties for each resource are specified
- Ensures that brand requirements are met

For more information about the `zonecfg` command, see the `zonecfg(1M)` man page.

### Branded Zone Configuration Components

This section covers the following components:

- Zone resources and properties that can be configured using the `zonecfg` command
- Resources included in the configuration by default

### Zone Name and Zone Path in an Branded Zone

You must choose a name and a path for your zone.
Zone Autoboot in an \texttt{lx} Branded Zone

The autoboot property setting determines whether the zone is automatically booted when the global zone is booted.

Resource Pool Association in an \texttt{lx} Branded Zone

If you have configured resource pools on your system as described in Chapter 13, "Creating and Administering Resource Pools (Tasks)," you can use the pool property to associate the zone with one of the resource pools when you configure the zone.

If you do not have resource pools configured, you can still specify that a subset of the system’s processors be dedicated to a non-global zone while it is running by using the dedicated-cpu resource. The system will dynamically create a temporary pool for use while the zone is running.

\textbf{Note} – A zone configuration using a persistent pool set through the pool property is incompatible with a temporary pool configured through the dedicated-cpu resource. You can set only one of these two properties.

Specifying the dedicated-cpu Resource

The dedicated-cpu resource specifies that a subset of the system's processors should be dedicated to a non-global zone while it is running. When the zone boots, the system will dynamically create a temporary pool for use while the zone is running.

Note that with specification in \texttt{zonecfg}, pool settings propagate during migrations.

The dedicated-cpu resource sets limits for ncpus, and optionally, importance.

\begin{itemize}
  \item \texttt{ncpus} Specify the number of CPUs or specify a range, such as 2–4 CPUs. If you specify a range because you want dynamic resource pool behavior, also do the following:
    \begin{itemize}
      \item Set the \texttt{importance} property.
      \item Enable the dynamic resource pool service as described in “Enabling and Disabling the Pools Facility” on page 167
    \end{itemize}
  \item \texttt{importance} If you are using a CPU range to achieve dynamic behavior, also set the \texttt{importance} property. The \texttt{importance} property, which is \texttt{optional}, defines the relative importance of the pool. This property is only needed when you specify a range for \texttt{ncpus} and are using dynamic resource pools managed by \texttt{poold}. If \texttt{poold} is not running, then \texttt{importance} is ignored. If \texttt{poold} is running and \texttt{importance} is not set, \texttt{importance} defaults to 1. For more information, see
\end{itemize}
“pool.importance Property Constraint” on page 153.

Caution – The cpu-shares rctl and the dedicated-cpu resource are incompatible.

Specifying the capped-cpu Resource

The capped-cpu resource provides an absolute limit on the amount of CPU resources that can be consumed by a project or a zone. The capped-cpu resource has a single ncpus property that is a positive decimal with two digits to the right of the decimal. This property corresponds to units of CPUs. The resource does not accept a range. The resource does accept a decimal number. When specifying ncpus, a value of 1 means 100 percent of a CPU. A value of 1.25 means 125 percent, because 100 percent corresponds to one full CPU on the system.

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

Scheduling Class in a Zone

You can use the fair share scheduler (FSS) to control the allocation of available CPU resources among zones, based on the importance of the workloads in the zone. This importance is expressed by the number of shares of CPU resources that you assign to each zone. Even if you are not using FSS to manage CPU resource allocation between zones, you can set the zone’s scheduling-class to use FSS so that you can set shares on projects within the zone.

When you explicitly set the cpu-shares property, the fair share scheduler (FSS) will be used as the scheduling class for that zone. However, the preferred way to use FSS in this case is to set FSS to be the system default scheduling class with the dispadm command. That way, all zones will benefit from getting a fair share of the system CPU resources. If cpu-shares is not set for a zone, the zone will use the system default scheduling class. The following actions set the scheduling class for a zone:

- You can use the scheduling-class property in zonecfg to set the scheduling class for the zone.
- You can set the scheduling class for a zone through the resource pools facility. If the zone is associated with a pool that has its pool.scheduler property set to a valid scheduling class, then processes running in the zone run in that scheduling class by default. See “Introduction to Resource Pools” on page 144 and “How to Associate a Pool With a Scheduling Class” on page 175.
- If the cpu-shares rctl is set and FSS has not been set as the scheduling class for the zone through another action, zoneadm sets the scheduling class to FSS when the zone boots.
If the scheduling class is not set through any other action, the zone inherits the system default scheduling class.

Note that you can use the `priocntl` described in the `priocntl(1)` man page to move running processes into a different scheduling class without changing the default scheduling class and rebooting.

### capped-memory Resource

The `capped-memory` resource sets limits for physical, swap, and locked memory. Each limit is optional, but at least one must be set.

- Determine values for this resource if you plan to cap memory for the zone by using `rcapd` from the global zone. The `physical` property of the `capped-memory` resource is used by `rcapd` as the `max-rss` value for the zone.
- The `swap` property of the `capped-memory` resource is the preferred way to set the `zone.max-swap` resource control.
- The `locked` property of the `capped-memory` resource is the preferred way to set the `zone.max-locked-memory` resource control.

**Note** – Applications generally do not lock significant amounts of memory, but you might decide to set locked memory if the zone’s applications are known to lock memory. If zone trust is a concern, you can also consider setting the locked memory cap to 10 percent of the system’s physical memory, or 10 percent of the zone’s physical memory cap.

For more information, see Chapter 10, “Physical Memory Control Using the Resource Capping Daemon (Overview),” Chapter 11, “Administering the Resource Capping Daemon (Tasks),” and “How to Configure the \textit{\textbackslash l}x Branded Zone” on page 440.

### Zone Network Interfaces in an \textit{\textbackslash l}x Branded Zone

Only the shared-IP network configuration is supported in an \textit{\textbackslash l}x branded zone.

Each zone that requires network connectivity must have one or more dedicated IP addresses. These addresses are associated with logical network interfaces. Network interfaces configured by the `zonecfg` command will automatically be set up and placed in the zone when it is booted.
Mounted File Systems in an \textit{\texttt{\textbackslash l\texttt{x}}} Branded Zone

Generally, the file systems mounted in a zone include the following:

- The set of file systems mounted when the virtual platform is initialized
- The set of file systems mounted from within the zone itself

This can include, for example, the following file systems:

- \texttt{automount}-triggered mounts
- Mounts explicitly performed by a zone administrator

Certain restrictions are placed on mounts performed from within the application environment. These restrictions prevent the zone administrator from denying service to the rest of the system, or otherwise negatively impacting other zones.

There are security restrictions associated with mounting certain file systems from within a zone. Other file systems exhibit special behavior when mounted in a zone. See “File Systems and Non-Global Zones” on page 353 for more information.

Zone-Wide Resource Controls in an \textit{\texttt{\textbackslash l\texttt{x}}} Branded Zone

The preferred, simpler method for setting a zone-wide resource control is to use the property name instead of the \texttt{\texttt{rct\texttt{l}}} resource. These limits are specified for both the global and non-global zones.

The global administrator can also set privileged zone-wide resource controls for a zone by using the \texttt{\texttt{rct\texttt{l}}} resource.

Zone-wide resource controls limit the total resource usage of all process entities within a zone. These limits are specified for both the global and non-global zones by using the \texttt{zonecfg} command. For instructions, see “How to Configure the \texttt{\texttt{\textbackslash l\texttt{x}}} Branded Zone” on page 440.

The following resource controls are currently available:
### TABLE 30–1 Zone-Wide Resource Controls

<table>
<thead>
<tr>
<th>Control Name</th>
<th>Global Property Name</th>
<th>Description</th>
<th>Default Unit</th>
<th>Value Used For</th>
<th>Locked Property of capped-memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone.cpu-cap</td>
<td></td>
<td>Absolute limit on the amount of CPU resources for this zone. A value of 100 means 100 percent of one CPU as the project.cpu-cap setting. A value of 125 is 125 percent, because 100 percent corresponds to one full CPU on the system when using CPU caps.</td>
<td>Quantity (number of CPUs)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zone.cpu-shares</td>
<td>cpu-shares</td>
<td>Number of fair share scheduler (FSS) CPU shares for this zone.</td>
<td>Quantity (shares)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zone.max-locked-memory</td>
<td></td>
<td>Total amount of physical locked memory available to a zone. If the privilege priv_proc_lock_memory is assigned to a zone, consider setting this resource control as well, to prevent that zone from locking all memory.</td>
<td>Size (bytes)</td>
<td></td>
<td>locked property of capped-memory</td>
</tr>
<tr>
<td>zone.max-lwps</td>
<td>max-lwps</td>
<td>Maximum number of LWPs simultaneously available to this zone.</td>
<td>Quantity (LWPs)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zone.max-msg-ids</td>
<td>max-msg-ids</td>
<td>Maximum number of message queue IDs allowed for this zone.</td>
<td>Quantity (message queue IDs)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zone.max-sem-ids</td>
<td>max-sem-ids</td>
<td>Maximum number of semaphore IDs allowed for this zone.</td>
<td>Quantity (semaphore IDs)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zone.max-shm-ids</td>
<td>max-shm-ids</td>
<td>Maximum number of shared memory IDs allowed for this zone.</td>
<td>Quantity (shared memory IDs)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Configurable Privileges in an lx Branded Zone

The `limitpriv` property is used to specify a privilege mask other than the predefined default set. When a zone is booted, a default set of privileges is included in the brand configuration. These privileges are considered safe because they prevent a privileged process in the zone from affecting processes in other non-global zones on the system or in the global zone. You can use the `limitpriv` property to do the following:

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

**Note** – There are a few privileges that cannot be removed from the zone’s default privilege set, and there are also a few privileges that cannot be added to the set at this time.

For more information, see "Privileges Defined in lx Branded Zones" on page 429, "Privileges in a Non-Global Zone" on page 367 and `privileges(5)`.

### attr Resource in an lx Branded Zone

You can use the `attr` resource type to enable access to an audio device present in the global zone. For instructions, see Step 12 of “How to Configure, Verify, and Commit the lx Branded Zone” on page 441.

You can also add a comment for a zone by using the `attr` resource type.
Resources Included in the Configuration by Default

Configured Devices in \(lx\) Branded Zones

The devices supported by each zone are documented in the man pages and other documentation for that brand. The \(lx\) zone does not allow the addition of any unsupported or unrecognized devices. The framework detects any attempt to add an unsupported device. An error message is issued that indicates the zone configuration cannot be verified.

Note that access to an audio device running in the global zone can be added through the \(attr\) resource property as shown in Step 12 of “How to Configure, Verify, and Commit the \(lx\) Branded Zone” on page 441.

File Systems Defined in \(lx\) Branded Zones

The file systems that are required for a branded zone are defined in the brand. You can add additional Solaris file systems to an \(lx\) branded zone by using the \(fs\) resource property as shown in Step 9 of “How to Configure, Verify, and Commit the \(lx\) Branded Zone” on page 441.

Note – Adding local Linux file systems is not supported. You can NFS mount file systems from a Linux server.

Privileges Defined in \(lx\) Branded Zones

Processes are restricted to a subset of privileges. Privilege restriction prevents a zone from performing operations that might affect other zones. The set of privileges limits the capabilities of privileged users within the zone.

Default, required default, optional, and prohibited privileges are defined by each brand. You can also add or remove certain privileges by using the \(limitpriv\) property as shown in Step 8 of “How to Configure, Verify, and Commit the \(lx\) Branded Zone” on page 441. The table Table 26–1 lists all of the Solaris privileges and the status of each privilege with respect to zones.

For more information about privileges, see the \(ppriv(1)\) man page and System Administration Guide: Security Services.
Using the zonecfg Command to Create an lx Branded Zone

The zonecfg command, which is described in the zonecfg(1M) man page, is used to configure a zone.

The zonecfg command can also be used to persistently specify the resource management settings for the global zone. For example, you can use the command to configure the global zone to use a dedicated CPU by using the dedicated-cpu resource.

The zonecfg command can be used in interactive mode, in command-line mode, or in command-file mode. The following operations can be performed using this command:

- Create or delete (destroy) a zone configuration
- Add resources to a particular configuration
- Set properties for resources added to a configuration
- Remove resources from a particular configuration
- Query or verify a configuration
- Commit to a configuration
- Revert to a previous configuration
- Rename a zone
- Exit from a zonecfg session

The zonecfg prompt is of the following form:

zonecfg:zonename>

When you are configuring a specific resource type, such as a file system, that resource type is also included in the prompt:

zonecfg:zonename:fs>

For more information, including procedures that show how to use the various zonecfg components described in this chapter, see "How to Configure the lx Branded Zone" on page 440.

zonecfg Modes

The concept of a scope is used for the user interface. The scope can be either global or resource specific. The default scope is global.

In the global scope, the add subcommand and the select subcommand are used to select a specific resource. The scope then changes to that resource type.

- For the add subcommand, the end or cancel subcommands are used to complete the resource specification.
For the select subcommand, the end or cancel subcommands are used to complete the resource modification.

The scope then reverts back to global.

Certain subcommands, such as add, remove, and set, have different semantics in each scope.

**zonecfg Interactive Mode**

In interactive mode, the following subcommands are supported. For detailed information about semantics and options used with the subcommands, see the `zonecfg(1M)` man page for options. For any subcommand that could result in destructive actions or loss of work, the system requests user confirmation before proceeding. You can use the `-F` (force) option to bypass this confirmation.

**help**
Print general help, or display help about a given resource.

```
zonecfg:lx-zone:net> help
```

**create**
Begin configuring an in-memory configuration for the specified new branded zone.

- With the `-t template` option, to create a configuration that is identical to the specified template. The zone name is changed from the template name to the new zone name. To create a Linux branded zone, use:

```
zonecfg:lx-zone> create -t SUNWlx
```

- With the `-b` option, to create a blank configuration for which you can set the brand.

```
zonecfg:lx-zone> create -b
zonecfg:lx-zone> set brand=lx
```

- With the `-F` option, to overwrite an existing configuration.

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

**add**
In the global scope, add the specified resource type to the configuration.

In the resource scope, add a property of the given name with the given value.

See How to Configure the lx Branded Zone and the `zonecfg(1M)` man page for more information.

**set**
Set a given property name to the given property value. Note that some properties, such as `zonepath`, are global, while others are resource specific. Thus, this command is applicable in both the global and resource scopes.
**zonecfg Modes**

**select**  
Applicable only in the global scope. Select the resource of the given type that matches the given property name-property value pair criteria for modification. The scope is changed to that resource type. You must specify a sufficient number of property name-value pairs for the resource to be uniquely identified.

**clear**  
Clear the value for optional settings. Required settings cannot be cleared. However, some required settings can be changed by assigning a new value.

**remove**  
In the global scope, remove the specified resource type. You must specify a sufficient number of property name-value pairs for the resource type to be uniquely identified. If no property name-value pairs are specified, all instances will be removed. If more than one exists, a confirmation is required unless the -F option is used.

In the resource scope, remove the specified property name-property value from the current resource.

**end**  
Applicable only in the resource scope. End the resource specification.

The zonecfg command then verifies that the current resource is fully specified.
- If the resource is fully specified, it is added to the in-memory configuration and the scope will revert back to global.
- If the specification is incomplete, the system displays an error message that describes what needs to be done.

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

**delete**  
Destroy the specified configuration. Delete the configuration both from memory and from stable storage. You must use the -F (force) option with delete.

**Caution** – This action is instantaneous. No commit is required, and a deleted zone cannot be reverted.

**info**  
Display information about the current configuration or the global resource properties zonepath, autoboot, and pool. If a resource type is specified, display information only about resources of that type. In the resource scope, this subcommand applies only to the resource being added or modified.

**verify**  
Verify current configuration for correctness. Ensure that all resources have all of their required properties specified.

**commit**  
Commit current configuration from memory to stable storage. Until the in-memory configuration is committed, changes can be removed with the revert subcommand. A configuration must be committed to be used by zoneadm. This
operation is attempted automatically when you complete a zonecfg session. Because only a correct configuration can be committed, the commit operation automatically does a verify.

revert
Revert configuration back to the last committed state.
exit
Exit the zonecfg session. You can use the -F (force) option with exit.
A commit is automatically attempted if needed. Note that an EOF character can also be used to exit the session.

zonecfg Command-File Mode
In command-file mode, input is taken from a file. The export subcommand described in zonecfg Interactive Mode is used to produce this file. The configuration can be printed to standard output, or the -f option can be used to specify an output file.

Branded Zone Configuration Data
Zone configuration data consists of two kinds of entities: resources and properties. Each resource has a type, and each resource can also have a set of one or more properties. The properties have names and values. The set of properties is dependent on the resource type.

Resource and Property Types
The resource and property types are described as follows:

Zone name
The zone name identifies the zone to the configuration utility. The following rules apply to zone names:
- Each zone must have a unique name.
- A zone name is case-sensitive.
- A zone name must begin with an alphanumeric character.

The name can contain alphanumeric characters, underbars (_), hyphens (-), and periods (.)
- The name cannot be longer than 64 characters.
- The name global and all names beginning with SUNW are reserved and cannot be used.

zonepath
The zonepath property is the path to the zone root. Each zone has a path to its root directory that is relative to the global zone’s root
directory. At installation time, the global zone directory is required to have restricted visibility. It must be owned by root with the mode 700.

The non-global zone’s root path is one level lower. The zone’s root directory has the same ownership and permissions as the root directory (/) in the global zone. The zone directory must be owned by root with the mode 755. These directories are created automatically with the correct permissions, and do not need to be verified by the zone administrator. This hierarchy ensures that unprivileged users in the global zone are prevented from traversing a non-global zone’s file system.

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home/export/lx-zone</td>
<td>zonecfg zonepath</td>
</tr>
<tr>
<td>/home/export/lx-zone/root</td>
<td>Root of the zone</td>
</tr>
<tr>
<td>/home/export/lx-zone/root/dev</td>
<td>Devices created for the zone</td>
</tr>
</tbody>
</table>

See “Traversing File Systems” on page 358 for a further discussion of this issue.

**Note** – You can move a zone to another location on the same system by specifying a new, full zonepath with the move subcommand of zoneadm. See “Moving a Non-Global Zone” on page 305 for instructions.

**autoboot**

If this property is set to true, the zone is automatically booted when the global zone is booted. Note that if the zones service, svc:/system/zones:default is disabled, the zone will not autoboot, regardless of the setting of this property. You can enable the zones service with the svcadm command described in the svcadm(1M) man page:

```
global# svcadm enable zones
```

**bootargs**

This property is used to set a boot argument for the zone. The boot argument is applied unless overridden by the reboot, zoneadm boot, or zoneadm reboot commands. See “Branded Zone Boot Arguments” on page 452.

**pool**

This property is used to associate the zone with a specific resource pool on the system. Multiple zones can share the resources of one pool. Also see “Specifying the dedicated-cpu Resource” on page 423.
limitpriv
This property is used to specify a privilege mask other than the default. See “Privileges in a Non-Global Zone” on page 367.

Privileges are added by specifying the privilege name, with or without the leading priv_. Privileges are excluded by preceding the name with a dash (-) or an exclamation mark (!). The privilege values are separated by commas and placed within quotation marks (“”).

As described in priv_str_to_set(3C), the special privilege sets of none, all, and basic expand to their normal definitions. Because zone configuration takes place from the global zone, the special privilege set zone cannot be used. Because a common use is to alter the default privilege set by adding or removing certain privileges, the special set default maps to the default, set of privileges. When default appears at the beginning of the limitpriv property, it expands to the default set.

The following entry adds the ability to set the system clock and removes the ability to send raw Internet Control Message Protocol (ICMP) packets:

global# zonecfg -z userzone
zonecfg:userzone> set limitpriv="default,sys_time,!net_icmpaccess"

If the zone’s privilege set contains a disallowed privilege, is missing a required privilege, or includes an unknown privilege, an attempt to verify, ready, or boot the zone will fail with an error message.

scheduling-class
This property sets the scheduling class for the zone. See “Scheduling Class in a Zone” on page 424 for additional information and tips.

dedicated-cpu
This resource dedicates a subset of the system’s processors to the zone while it is running. The dedicated-cpu resource provides limits for ncpus and, optionally, importance. For more information, see “Specifying the dedicated-cpu Resource” on page 423.

capped-cpu
This resource establishes an absolute limit on the number of CPUs for this zone. The capped-cpu resource provides limits for ncpus. For more information, see “Specifying the capped-cpu Resource” on page 424.

capped-memory
This resource groups the properties used when capping memory for the zone. The capped-memory resource provides limits for physical, swap, and locked memory. At least one of these properties must be specified.

fs
Each zone can have various file systems that are mounted when the zone transitions from the installed state to the ready state. The file system resource specifies the path to the file system mount point. For
more information about the use of file systems in zones, see “File Systems and Non-Global Zones” on page 353.

net

The network interface resource is the interface name. Each zone can have network interfaces that are be set up when the zone transitions from the installed state to the ready state.

Only the shared-IP network configuration is supported in an `lx` branded zone

rctl

The `rctl` resource is used for zone-wide resource controls. The controls are enabled when the zone transitions from the installed state to the ready state.

**Note** – To configure zone-wide controls using the `set global_property_name` subcommand of `zonecfg` instead of the `rctl` resource, see “How to Configure the `lx` Branded Zone” on page 440.

attr

This generic attribute can be used for user comments or by other subsystems. The `name` property of an `attr` must begin with an alphanumeric character. The `name` property can contain alphanumeric characters, hyphens (`-`), and periods (`.`). Attribute names beginning with `zone.` are reserved for use by the system.

### Resource Type Properties in the `lx` Branded Zone

Resources also have properties to configure. The following properties are associated with the resource types shown.

**dedicated-cpu** `ncpus, importance`

Specify the number of CPUs and, optionally, the relative importance of the pool. The following example specifies a CPU range for use by the zone `lx-zone`. `importance` is also set.

```
zonecfg:lx-zone> add dedicated-cpu
zonecfg:lx-zone:dedicated-cpu> set ncpus=1-3
zonecfg:lx-zone:dedicated-cpu> set importance=2
zonecfg:lx-zone:dedicated-cpu> end
```

**capped-cpu** `ncpus`

Specify the number of CPUs. The following example specifies a CPU limit of 3.5 CPUs for use by the zone `lx-zone`.
capped-memory  physical, swap, locked

This resource groups the properties used when capping memory for the zone. The following example specifies the memory limits for the zone lx-zone. Each limit is optional, but at least one must be set.

```plaintext
zonecfg:lx-zone> add capped-memory
zonecfg:lx-zone:capped-memory> set =50m
zonecfg:lx-zone:capped-memory> set swap=100m
zonecfg:lx-zone:capped-memory> set locked=30m
zonecfg:lx-zone:capped-memory> end
```

fs  dir, special, raw, type, options

The lines in the following example add read-only access to CD or DVD media in a non-global zone. The file system is loopback mounted with the options ro, nodevices (read-only and no devices) in the non-global zone.

```plaintext
zonecfg:lx-zone> add fs
zonecfg:lx-zone:fs> set dir=/cdrom
zonecfg:lx-zone:fs> set special=/cdrom
zonecfg:lx-zone:fs> set type=lofs
zonecfg:lx-zone:fs> add options [ro,nodevices]
zonecfg:lx-zone:fs> end
```

Note that section 1M man pages are available for mount options that are unique to a specific file system. The names of these man pages have the form mount_filesystem.

net  address, physical

In the following example, IP address 192.168.0.1 is added to a zone. An bge0 card is used for the physical interface.

```plaintext
zonecfg:lx-zone> add net
zonecfg:lx-zone:net> set physical=bge0
zonecfg:lx-zone:net> set address=192.168.0.1
zonecfg:lx-zone:net> end
```
Note – To determine which physical interface to use, type `ifconfig -a` on your system. Each line of the output, other than loopback driver lines, begins with the name of a card installed on your system. Lines that contain `LOOPBACK` in the descriptions do not apply to cards.

Available zone-wide resource controls are described in “Zone-Wide Resource Controls in an \lx Branded Zone” on page 426.

In the following example, a comment about a zone is added.

```
zonecfg:\lx-zone> add attr
zonecfg:\lx-zone:attr> set name=comment
zonecfg:\lx-zone:attr> set type=string
zonecfg:\lx-zone:attr> set value="Production zone"
zonecfg:\lx-zone:attr> end
```

You can use the `export` subcommand to print a zone configuration to standard output. The configuration is saved in a form that can be used in a command file.
Configuring the lx Branded Zone (Tasks)

This chapter describes how to configure an lx branded zone on your x64 or x86 based system. The process is basically the same as the procedure to configure a Solaris Zone. A few of the properties are not needed to configure a branded zone.

Planning and Configuring an lx Branded Zone (Task Map)

Before you set up your system to use zones, you must first collect information and make decisions about how to configure the zones. The following task map summarizes how to plan and configure an lx zone.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan your zone strategy.</td>
<td>■ Determine which applications you want to run in zones.</td>
<td>See “System and Space Requirements” on page 421 and “Resource Pools Used in Zones” on page 145.</td>
</tr>
<tr>
<td></td>
<td>■ Assess the availability of disk space to hold the files in the zone.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ If you are also using resource management features, determine how to align the zone with the resource management boundaries.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ If you are using resource pools, configure the pools if necessary.</td>
<td></td>
</tr>
</tbody>
</table>
### Task Description

**Task**

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine the name and the path for the zone.</td>
<td>Decide what to call the zone based on the naming conventions. A path on a Zetabyte File System (ZFS) is recommended. When the source zonepath and the target zonepath both reside on ZFS and are in the same pool, the zoneadm clone command automatically uses ZFS to clone the zone.</td>
<td>See “Resource and Property Types” on page 433 and Solaris ZFS Administration Guide.</td>
</tr>
<tr>
<td>Obtain or configure IP addresses for the zone.</td>
<td>Depending on your configuration, you must obtain at least one IP address for each non-global zone that you want to have network access.</td>
<td>See “Determine the Zone Host Name and the Network Requirements” on page 246 and System Administration Guide: IP Services.</td>
</tr>
<tr>
<td>Determine if you want to mount file systems in the zone.</td>
<td>Review your application requirements.</td>
<td>See “File Systems Mounted in Zones” on page 221 for more information.</td>
</tr>
<tr>
<td>Determine which network interfaces should be made available in the zone.</td>
<td>Review your application requirements.</td>
<td>See “Shared-IP Network Interfaces” on page 361 for more information.</td>
</tr>
<tr>
<td>Determine whether you must alter the default set of non-global zone permissions.</td>
<td>Check the set of privileges: default, privileges that can be added and removed, and privileges that cannot be used at this time.</td>
<td>See “Resource and Property Types” on page 433 and “Privileges in a Non-Global Zone” on page 367.</td>
</tr>
<tr>
<td>Configure the zone.</td>
<td>Use zonecfg to create a configuration for the zone.</td>
<td>See “How to Configure, Verify, and Commit the lx Branded Zone” on page 441.</td>
</tr>
<tr>
<td>Verify and commit the configured zone.</td>
<td>Determine whether the resources and properties specified are valid on a hypothetical system.</td>
<td>See “How to Configure, Verify, and Commit the lx Branded Zone” on page 441.</td>
</tr>
</tbody>
</table>

### How to Configure the lx Branded Zone

You use the zonecfg command described in the zonecfg(1M) man page to perform the following actions.

- Create the zone configuration
- Verify that all required information is present
- Commit the non-global zone configuration
Tip – If you know you will be using CDs or DVDs to install applications in an \( \text{lx} \) branded zone, use \texttt{add fs} to add read-only access to CD or DVD media in the global zone when you initially configure the branded zone. A CD or DVD can then be used to install a product in the branded zone.

While configuring a zone with the \texttt{zonectfg} utility, you can use the \texttt{revert} subcommand to undo the setting for a resource. See “How to Revert a Zone Configuration” on page 261.

A script to configure multiple zones on your system is provided in “Script to Configure Multiple \( \text{lx} \) Branded Zones” on page 445.

To display a non-global zone’s configuration, see “How to Display the Configuration of a Branded Zone” on page 447.

Tip – After you have configured the branded zone, it is a good idea to make a copy of the zone’s configuration. You can use this backup to restore the zone in the future. As superuser or Primary Administrator, print the configuration for the zone \( \text{lx} \)-zone to a file. This example uses a file named \( \text{lx} \)-zone\.config.

\begin{verbatim}
global# zonectfg \text{-z} \text{lx-zone export > lx-zone.config}
\end{verbatim}

See “How to Restore an Individual Non-Global Zone” on page 402 for more information.

\section*{How to Configure, Verify, and Commit the \( \text{lx} \) Branded Zone}

Note that you cannot use \( \text{lx} \) branded zones on a Trusted Solaris system where labels are enabled. The \texttt{zoneadm} command will not verify the configuration.

You must be the global administrator in the global zone to perform this procedure.

1 \textbf{Become superuser, or assume the Primary Administrator role.}

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in \textit{System Administration Guide: Basic Administration}.

2 \textbf{Set up a zone configuration with the zone name you have chosen.}

The name \( \text{lx-zone} \) is used in this example procedure.

\begin{verbatim}
  global# zonectfg \text{-z} \text{lx-zone}
\end{verbatim}
If this is the first time you have configured this zone, you will see the following system message:

```
lx-zone: No such zone configured
Use 'create' to begin configuring a new zone.
```

3 **Create the new `lx` zone configuration by using the SUNWlx template.**

```
zonecfg:lx-zone> create -t SUNWlx
```

Alternatively, you can create a blank zone and explicitly set the brand:

```
zonecfg:lx-zone> create -b
zonecfg:lx-zone> set brand=lx
```

4 **Set the zone path, `/export/home/lx-zone` in this procedure.**

```
zonecfg:lx-zone> set zonepath=/export/home/lx-zone
```

5 **Set the autoboot value.**

If set to `true`, the zone is automatically booted when the global zone is booted. Note that for the zones to autoboot, the zones service `svc:/system/zones:default` must also be enabled. The default value is `false`.

```
zonecfg:lx-zone> set autoboot=true
```

6 **Set persistent boot arguments for a zone.**

```
zonecfg:lx-zone> set bootargs="-i=altinit"
```

7 **If resource pools are enabled on your system, associate a pool with the zone.**

This example uses the default pool, named `pool_default`.

```
zonecfg:lx-zone> set pool=pool_default
```

Because a resource pool can have an optional scheduling class assignment, you can use the pools facility to set a default scheduler other than the system default for a non-global zone. For instructions, see “How to Associate a Pool With a Scheduling Class” on page 175 and “Creating the Configuration” on page 190.

8 **Revise the default set of privileges.**

```
zonecfg:lx-zone> set limitpriv="default,proc_prioctrl"
```

The `proc_prioctrl` privilege is used to run processes in the real-time class.

9 **Set five CPU shares.**

```
zonecfg:lx-zone> set cpu-shares=5
```
10 Add a memory cap.
zonecfg:lx-zone> add capped-memory

a. Set the memory cap.
zonecfg:lx-zone:capped-memory> set =50m

b. Set the swap memory cap.
zonecfg:lx-zone:capped-memory> set swap=100m

c. Set the locked memory cap.
zonecfg:lx-zone:capped-memory> set locked=30m

d. End the specification.
zonecfg:lx-zone:capped-memory> end

11 Add a file system.
zonecfg:lx-zone> add fs

a. Set the mount point for the file system, /export/linux/local in this procedure.
zonecfg:lx-zone:fs> set dir=/export/linux/local

b. Specify that /opt/local in the global zone is to be mounted as /export/linux/local in the zone being configured.
zonecfg:lx-zone:fs> set special=/opt/local

In the non-global zone, the /export/linux/local file system will be readable and writable.

c. Specify the file system type, lofs in this procedure.
zonecfg:lx-zone:fs> set type=lofs

The type indicates how the kernel interacts with the file system.

d. End the file system specification.
zonecfg:lx-zone:fs> end

This step can be performed more than once to add more than one file system.

12 Add a network interface.
zonecfg:lx-zone> add net

a. Set the IP address in the form ip address of zone/netmask. In this procedure, 10.6.10.233/24 is used.
zonecfg:lx-zone:net> set address=10.6.10.233/24
b. Set the physical device type for the network interface, the bge device in this procedure.
   
   zonecfg:lx-zone:net> set physical=bge

   zonecfg:lx-zone:net> set physical=bge0

c. (Optional) Set the default router for the network interface, 10.0.0.1 in this procedure.
   
   zonecfg:my-zone:net> set defrouter=10.0.0.1

d. End the specification.
   
   zonecfg:lx-zone:net> end

   This step can be performed more than once to add more than one network interface.

13 Enable an audio device present in the global zone in this zone by using the attr resource type.
   
   zonecfg:lx-zone> add attr

   a. Set the name to audio.
      
      zonecfg:lx-zone:attr> set name=audio

   b. Set the type to boolean.
      
      zonecfg:lx-zone:attr> set type=boolean

   c. Set the value to true.
      
      zonecfg:lx-zone:attr> set value=true

   d. End the attr resource type specification.
      
      zonecfg:lx-zone:attr> end

14 Verify the zone configuration for the zone.
   
   zonecfg:lx-zone> verify

15 Commit the zone configuration for the zone.
   
   zonecfg:lx-zone> commit

16 Exit the zonecfg command.
   
   zonecfg:lx-zone> exit

   Note that even if you did not explicitly type commit at the prompt, a commit is automatically attempted when you type exit or an EOF occurs.
Using Multiple Subcommands From the Command Line

Tip – The zonecfg command also supports multiple subcommands, quoted and separated by semicolons, from the same shell invocation.

```sh
global# zonecfg -z lx-zone "create -t SUNWlx; set zonepath=/export/home/lx-zone"
```

Where to Go From Here

See Installing and Booting lx Branded Zones “Installing and Booting Zones” on page 274 to install your committed zone configuration.

Script to Configure Multiple lx Branded Zones

You can use this script to configure and boot multiple zones on your system. The script takes the following parameters:

- The number of zones to be created
- The zonename prefix
- The directory to use as the base directory

You must be the global administrator in the global zone to execute the script. The global administrator has superuser privileges in the global zone or assumes the Primary Administrator role.

```sh
#!/bin/ksh
#
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
#ident %Z%%M% %I% %E% SMI
if [[ -z "$1" || -z "$2" || -z "$3" || -z "$4" ]]; then
    echo "usage: $0 <#-of-zones> <zonename-prefix> <basedir> <template zone>"
    exit 2
fi
if [[ ! -d "$3" ]]; then
    echo "$3 is not a directory"
    exit 1
fi
state=`zoneadm -z "$4" list -p 2>/dev/null | cut -f 3 -d ":"`
if [[ -z "$state" || $state != "installed" ]]; then
    echo "$4 must be an installed, halted zone"
    exit 1
fi
```
template_zone=$4

nprocs=`psrinfo | wc -l`
nzones=$1
prefix=$2
dir=$3

ip_addrs_per_if='ndd /dev/ip ip_addrs_per_if'
if [ $ip_addrs_per_if -lt $nzones ]; then
    echo "ndd parameter ip_addrs_per_if is too low ($ip_addrs_per_if)"
    echo "set it higher with 'ndd -set /dev/ip ip_addrs_per_if <num>"
    exit 1
fi

i=1
while [ $i -le $nzones ]; do
    zoneadm -z $prefix$i clone $template_zone > /dev/null 2>&1
    if [ $? != 0 ]; then
        echo configuring $prefix$i
        F=$dir/$prefix$i.config
        rm -f $F
        echo "create -t SUNWlx" >> $F
        echo "set zonepath=$dir/$prefix$i" >> $F
        zonecfg -z $prefix$i -f $dir/$prefix$i.config 2>&1 | sed 's/^/ /g'
    else
        echo "skipping $prefix$i, already configured"
    fi
    i=`expr $i + 1`
done

i=1
while [ $i -le $nzones ]; do
    j=1
    while [ $j -le $nprocs ]; do
        if [ $i -le $nzones ]; then
            if [ 'zoneadm -z $prefix$i list -p | \
                cut -d':' -f 3' != "configured" ]; then
                echo "skipping $prefix$i, already installed"
            else
                echo installing $prefix$i
                mkdir -pm 0700 $dir/$prefix$i
                chmod 700 $dir/$prefix$i
                zoneadm -z $prefix$i install -s -d /path/to/ISOs > /dev/null 2>&1 &
                sleep 1 # spread things out just a tad
            fi
        fi
        j=`expr $j + 1`
    done
    i=`expr $i + 1`
done

How to Configure the lx Branded Zone

System Administration Guide: Virtualization Using the Solaris Operating System • October 2009 (Beta)
```bash
i='expr $i + 1'

j='expr $j + 1'
done

wait
done

i=1
para='expr $nprocs \* 2'
while [ $i -le $nzones ]; do
date
j=1
while [ $j -le $para ]; do
    if [ $i -le $nzones ]; then
        echo booting $prefix$i
        zoneadm -z $prefix$i boot &
    fi
    j='expr $j + 1'
    i='expr $i + 1'
done
wait
done
```

▼ **How to Display the Configuration of a Branded Zone**

You must be the global administrator in the global zone to perform this procedure.

1 **Become superuser, or assume the Primary Administrator role.**

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 **Display the configuration of a zone.**

   ```bash
global# zonecfg -z zonename info
   ```

### Modifying, Reverting, or Removing Zone Configurations

The following sections contain procedures for modifying, reverting, or removing a zone configuration.

- “How to Modify a Resource Type in a Zone Configuration” on page 257
- “How to Clear a Property Type in a Zone Configuration” on page 258
- “How to Rename a Zone” on page 259
- “How to Revert a Zone Configuration” on page 261
- “How to Delete a Zone Configuration” on page 263
This chapter discusses the following topics:
- Installing an lx zone on your system
- Cloning a zone on your system
- Halting, rebooting, and uninstalling zones

Branded Zone Installation and Administration Concepts

The `zoneadm` command described in the `zoneadm(1M)` man page is the primary tool used to install and administer non-global zones. Operations using the `zoneadm` command must be run from the global zone. The following tasks can be performed using the `zoneadm` command:
- Verify a zone
- Install a zone
- Boot a zone
- Display information about a running zone
- Halt a zone
- Reboot a zone
- Uninstall a zone
- Relocate a zone from one point on a system to another point on the same system
- Provision a new zone based on the configuration of an existing zone on the same system
- Migrate a zone, used with the `zonecfg` command

For zone installation and verification procedures, see Chapter 33, “Installing, Booting, Halting, Uninstalling and Cloning lx Branded Zones (Tasks),” and the `zoneadm(1M)` man page. Also refer to the `zoneadm(1M)` man page for supported options to the `zoneadm list` command.
zone configuration procedures, see Chapter 31, “Configuring the \( \texttt{lx} \) Branded Zone (Tasks),” and the \texttt{zonecfg(1M)} man page. Zone states are described in “Non-Global Zone State Model” on page 207.

If you plan to produce Solaris auditing records for zones, read “Using Solaris Auditing in Zones” on page 372 before you install non-global zones.

\textbf{Note} – Once the zone is installed, all software configuration and management has to be done by the zone administrator using Linux tools from inside the zone.

\textbf{lx} \textbf{Branded Zone Installation Methods}

You can install an \( \texttt{lx} \) branded zone by using a tarball, CD-ROM or DVD discs, or an ISO image. If you install from discs or from an ISO image, you can specify Sun package cluster categories. The categories are cumulative. If you do not specify a cluster, the default is \texttt{desktop}.

\begin{table}[h]
\centering
\renewcommand{\arraystretch}{1.5}
\begin{tabular}{|l|p{12cm}|}
\hline
\textbf{Sun Category} & \textbf{Contents} \\
\hline
\texttt{core} & The minimum set of packages needed to construct a zone. \\
\hline
\texttt{server} & core plus server-oriented packages, such as \texttt{httpd}, \texttt{mailman}, \texttt{imapd}, and \texttt{spam-assassin}. \\
\hline
\texttt{desktop} & server plus user-oriented packages, such as \texttt{evolution}, \texttt{gimp}, \texttt{mozilla}, and \texttt{openoffice}. \\
\hline
\texttt{developer} & desktop plus developer packages, such as \texttt{bison}, \texttt{emacs}, \texttt{gcc}, \texttt{vim-x11}, and many library development packages. \\
\hline
\texttt{all} & Everything on the install media that is known not to interfere with the zone's operation. Certain packages might not function in a Linux zone. \\
\hline
\end{tabular}
\end{table}

To install configured \( \texttt{lx} \) branded zones, see “How to Install an \( \texttt{lx} \) Branded Zone” on page 456.
Branded Zone Construction

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

After you have configured a non-global zone, you should verify that the zone can be installed safely on your system's configuration. You can then install the zone. The files needed for the zone's root file system are installed by the system under the zone's root path. The Linux zone will be populated from CD, ISO images, or a tarball, as described in “How to Install an \x Branded Zone” on page 456.

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

A zone in the ready state does not have any user processes executing in it. The primary difference between a ready zone and a running zone is that at least one process is executing in a running zone. See the init(1M) man page for more information.

In the ready state, the zsch ed and zoneadm d processes are started to manage the virtual platform.

zoneadm d Zones Administration Daemon

The zones administration daemon, zoneadm d, is the primary process for managing the zone's virtual platform. For more information, see “The zoneadm d Daemon” on page 268.

zsched Zone Scheduling Process

The process that manages the application environment, zsch ed, is described in “The zsch ed Zone Scheduler” on page 268.

Branded Zone Application Environment

The zoneadm command is used to create the zone application environment.

All additional configuration is done by the zone administrator using Linux tools from within the zone.
Passwords

Note that the root (superuser) password will be root when the zone is installed from the Sun tarball. The root (superuser) password will be unset (blank) when the zone is installed from ISO images or a CD.

About Halting, Rebooting, Uninstalling, and Cloning Branded Zones

This section provides an overview of the procedures for halting, rebooting, uninstalling, and cloning zones.

Halting a Branded Zone

The `zoneadm halt` command is used to remove both the application environment and the virtual platform for a zone. The zone is then brought back to the installed state. All processes are killed, devices are unconfigured, network interfaces are destroyed, file systems are unmounted, and the kernel data structures are destroyed.

The `halt` command does not run any shutdown scripts within the zone. To shut down a zone, see “How to Use `zlogin` to Shut Down a Zone” on page 301.

If the halt operation fails, see “Zone Does Not Halt” on page 407.

Rebooting a Branded Zone

The `zoneadm reboot` command is used to reboot a zone. The zone is halted and then booted again. The zone ID will change when the zone is rebooted.

Branded Zone Boot Arguments

Zones support the following boot arguments used with the `zoneadm boot` and `reboot` commands:

- `-i altinit`
- `-s`

The following definitions apply:

- `-i altinit` Selects an alternative executable to be the first process. `altinit` must be a valid path to an executable. The default first process is described in `init(1M)`. 
Boots the zone to init level s.

For usage examples, see "How to Boot an lxl Branded Zone" on page 461 and "How to Boot an lxl Branded Zone in Single-User Mode" on page 462.

For information on the init command, see init(1M).

Branded Zone autoboot

If you set the autoboot resource property in a zone’s configuration to true, that zone is automatically booted when the global zone is booted. The default setting is false.

Note that for zones to autoboot, the zones service svc:/system/zones:default must also be enabled.

Uninstalling the Branded Zone

The zoneadm uninstall command removes all of the files under the zone’s root file system. Before proceeding, the command prompts you to confirm the action, unless the -F (force) option is also used. Use the uninstall command with caution, because the action is irreversible.

About Cloning an lxl Branded Zone

Cloning allows you to copy an existing configured and installed zone on your system to rapidly provision a new zone on the same system. For more information about the clone process, see “About Cloning Non-Global Zones” on page 271. To clone an lxl branded zone, see “Cloning an lxl Branded Zone on the Same System” on page 466.

Booting and Rebooting lxl Branded Zones

For procedures to boot and reboot zones, see “How to Boot an lxl Branded Zone” on page 461 and "How to Reboot an lxl Branded Zone” on page 464.
Installing, Booting, Halting, Uninstalling and Cloning lx Branded Zones (Tasks)

This chapter describes how to install and boot an lx branded zone. The following other tasks are also addressed:

- Using clone to install a zone on the same system
- Halting, rebooting, and uninstalling zones
- Removing a zone from a system

### lx Branded Zone Installation (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obtain the Linux archives.</td>
<td>Before you can install the lx branded zone, you must first obtain the Linux archives.</td>
<td>“How to Obtain the Linux Archives” on page 456</td>
</tr>
<tr>
<td>Install a configured lx branded zone.</td>
<td>Install a zone that is in the configured state.</td>
<td>“How to Install an lx Branded Zone” on page 456</td>
</tr>
<tr>
<td>(Optional) Install a subset of the available packages.</td>
<td>When installing from CD or ISO images, you can install a subset of the packages on the install media.</td>
<td>“How to Install a Subset of the Packages” on page 459</td>
</tr>
<tr>
<td>(Optional) Enable networking in the zone.</td>
<td>Networking is disabled by default and must be enabled if you want this functionality.</td>
<td>“How to Enable Networking in an lx Branded Zone” on page 459</td>
</tr>
<tr>
<td>Obtain the universally unique identifier (UUID) for the zone.</td>
<td>This separate identifier, assigned when the zone is installed, is an alternate way to identify a zone.</td>
<td>“How to Obtain the UUID of an Installed Branded Zone” on page 459</td>
</tr>
<tr>
<td>(Optional) Transition an installed zone to the ready state.</td>
<td>You can skip this procedure if you want to boot the zone and use it immediately.</td>
<td>“(Optional) Placing an Installed lxBranded Zone in the Ready State” on page 461</td>
</tr>
</tbody>
</table>
Installing and Booting \texttt{\textsc{lx}} Branded Zones

Use the zoneadm command described in the zoneadm(1M) man page to perform installation tasks for a non-global zone.

\section*{How to Obtain the Linux Archives}

Before you can install the \texttt{\textsc{lx}} branded zone, you must first obtain the Linux archives. The archives are distributed in the following forms:

\begin{itemize}
  \item A compressed \texttt{tar} archive (a \texttt{tarball})
  \item A set of CD-ROM or DVD discs
  \item A group of ISO images
\end{itemize}

\textbullet \quad \textbf{Obtain the Linux distribution using one of the following methods:}

\begin{itemize}
  \item To download a tarball, go to \url{http://opensolaris.org/os/community/brandz/downloads}. Follow the instructions on the downloads site.
  \item To obtain a set of CD-ROM or DVD discs, go to the CentOS site at \url{http://www.centos.org} or the Red Hat site at \url{http://www.redhat.com}.
  \item To obtain an ISO image, go to the CentOS site at \url{http://www.centos.org} or the Red Hat site at \url{http://www.redhat.com}.
\end{itemize}

\section*{How to Install an \texttt{\textsc{lx}} Branded Zone}

This procedure is used to install a configured \texttt{\textsc{lx}} branded zone. Once the zone is installed, all software configuration and management has to be done by the zone administrator using Linux tools from inside the zone.

See \texttt{Example 33–1}, \texttt{Example 33–2}, and \texttt{Example 33–3} for examples of zone installation command lines using the different distribution paths. If you install from discs or from an ISO image, you must specify Sun package cluster categories. See “\texttt{\textsc{lx} Branded Zone Installation Methods}” on page 450 for information on package cluster categories.
Note that you can verify a zone prior to installing it. If you skip this procedure, the verification is performed automatically when you install the zone. The procedure is documented in “(Optional) How to Verify a Configured Zone Before It Is Installed” on page 274.

You must be the global administrator in the global zone to perform this procedure.

---

**Note** – In Step 2, if the zonepath is on ZFS, the zoneadm install command automatically creates a ZFS file system (dataset) for the zonepath when the zone is installed. You can block this action by including the `-x nodataset` parameter.

---

1. **Become superuser, or assume the Primary Administrator role.**

   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. **(Optional) If you intend to install from DVD or CD, enable volfs on your system and verify that it is running.**

   ```
   global# svcadm enable svc:/system/filesystem/volfs:default
   ```

   ```
   global# svcs | grep volfs
   ```

   You will see a display similar to the following:

   ```
   online 17:30 svc:/system/filesystem/volfs:default
   ```

3. **Install the configured zone *lx-zone* by using the zoneadm command with the *install* option and the path to the archive.**

   - **Install the zone, automatically creating a ZFS file system if the zonepath is on ZFS.**

     ```
     global# zoneadm -z lx-zone install -d archive_path
     ```

     The system will display:

     ```
     A ZFS file system has been created for this zone.
     ```

   - **Install the zone that has a zonepath on ZFS, but do not automatically create the ZFS file system.**

     ```
     global# zoneadm -z lx-zone install -x nodataset -d archive_path
     ```

     You will see various messages as the files and directories needed for the zone’s root file system, as well as the package files, are installed under the zone’s root path.

---

**Note** – If you do not specify `archive_path`, the default is CD.
4  (Optional) If an error message is displayed and the zone fails to install, type the following to get the zone state:

```
global# zoneadm -z lx-zone list -iv
```

- If the state is listed as configured, make the corrections specified in the message and try the `zoneadm install` command again.
- If the state is listed as incomplete, first execute this command:

```
global# zoneadm -z lx-zone uninstall
```

Then make the corrections specified in the message, and try the `zoneadm install` command again.

5  When the installation completes, use the `list` subcommand with the `-i` and `-v` options to list the installed zones and verify the status.

```
global# zoneadm list -iv
```

You will see a display that is similar to the following:

```
ID  NAME  STATUS  PATH                  BRAND  IP
0   global running /                  native shared
   lx-zone installed /export/home/lx-zone lx shared
```

**Example 33–1**  Install Command Using a CentOS Compressed tar Archive

```
global# zoneadm -z lx-zone install -d /export/centos_fs_image.tar.bz2
```

**Example 33–2**  Install Command Using CentOS CDs

For CD or DVD installation, `volfs` must be enabled on your system. You must specify a software cluster package. For example, use `development` to install a full environment, or type the names of particular clusters. If you do not specify a cluster package, desktop is installed by default. The CD device is `/cdrom/cdrom0`.

```
global# zoneadm -z lx-zone install -d /cdrom/cdrom0 development
```

**Example 33–3**  Install Command Using CentOS ISO Images

You must specify a software cluster package. Use `development` to install a full environment, or specify particular clusters. If you do not specify a cluster package, desktop is installed by default. The CentOS ISO images reside in the directory `/export/centos_3.7`.

```
global# zoneadm -z lx-zone install -d /export/centos_3.7 development
```

**See Also**  For more information on datasets, see *Solaris ZFS Administration Guide*
Troubleshooting  If a zone installation is interrupted or fails, the zone is left in the incomplete state. Use `uninstall -F` to reset the zone to the configured state.

▼ How to Install a Subset of the Packages

When installing from CD or ISO images, you can install a subset of the packages on the install media. The available subsets are `core`, `server`, `desktop`, `developer`, and `all`.

You must be the global administrator in the global zone to perform this procedure.

1  **Become superuser, or assume the Primary Administrator role.**
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2  **Install only the server package:**
   
   ```
   global# zoneadm -z lx-zone install -d archive_path server
   ```

▼ How to Enable Networking in an `lx` Branded Zone

When you install an `lx` branded zone, networking is disabled. Use a procedure such as this one to enable networking.

You must be the zone administrator to perform this procedure.

1  **Edit the `/etc/sysconfig/network` file in the zone.**
   
   ```
   NETWORKING=yes
   HOSTNAME=your.hostname
   ```

2  **To set up a NIS domain, add a line similar to the following:**
   
   ```
   NISDOMAIN=domain.Sun.COM
   ```

More Information  Configuring Networking and Naming Services

For more information on configuring networking or naming services, consult the documentation for your Linux distribution.

▼ How to Obtain the UUID of an Installed Branded Zone

A universally unique identifier (UUID) is assigned to a zone when it is installed. The UUID can be obtained by using `zoneadm` with the `list` subcommand and the `-p` option. The UUID is the fifth field of the display.
View the UUIDs for zones that have been installed.

```
  global# zoneadm list -p
```

You will see a display similar to the following:

```
  0:global:running::native
  1:centos38:running::zones/centos38:27fabdc8-d8ce-e8aa-9921-ad1ea23ab063:lx
```

**Example 33–4** How to Use the UUID in a Command

```
  global# zoneadm -z lx-zone -u 61901255-35cf-40d6-d501-f37dc84eb504 list -v
```

If both `-u uuid-match` and `-z zonename` are present, the match is done based on the UUID first. If a zone with the specified UUID is found, that zone is used, and the `-z` parameter is ignored. If no zone with the specified UUID is found, then the system searches by the zone name.

**About the UUID**

Zones can be uninstalled and reinstalled under the same name with different contents. Zones can also be renamed without the contents being changed. For these reasons, the UUID is a more reliable handle than the zone name.

**See Also**

For more information, see `zoneadm(1M)` and `libuuid(3LIB)`.

**How to Mark an Installed lx Branded Zone Incomplete**

If administrative changes on the system have rendered a zone unusable or inconsistent, it is possible to change the state of an installed zone to incomplete.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**

   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Mark the zone testzone incomplete.**

   ```
   1 global# zoneadm -z testzone mark incomplete
   ```

3. **Use the list subcommand with the `-i` and `-v` options to verify the status.**

   ```
   1 global# zoneadm list -iv
   ```
You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td></td>
<td>testzone</td>
<td>incomplete</td>
<td>/export/home/testzone</td>
<td>lx</td>
<td>shared</td>
</tr>
</tbody>
</table>

**More Information**

**Marking a Zone Incomplete**

**Note** – Marking a zone incomplete is irreversible. The only action that can be taken on a zone marked incomplete is to uninstall the zone and return it to the configured state. See “How to Uninstall a Branded Zone” on page 465.

**(Optional) Placing an Installed lx Branded Zone in the Ready State**

Transitioning into the ready state prepares the virtual platform to begin running user processes. Zones in the ready state do not have any user processes executing in them.

You can skip this procedure if you want to boot the zone and use it immediately. The transition through the ready state is performed automatically when you boot the zone.

See “(Optional) How to Transition the Installed Zone to the Ready State” on page 278

**▼ How to Boot an lx Branded Zone**

Booting a zone places the zone in the running state. A zone can be booted from the ready state or from the installed state. A zone in the installed state that is booted transparently transitions through the ready state to the running state. Zone login is allowed for zones in the running state.

You must be the global administrator in the global zone to perform this procedure.

**Tip** – Note that you cannot boot an lx branded zone on a Solaris Trusted Extensions system that has labels enabled.

1. **Become superuser, or assume the Primary Administrator role.**

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.
2 Use the `zoneadm` command with the `-z` option, the name of the zone, which is `lx-zone`, and the `boot` subcommand to boot the zone.

```
global# zoneadm -z lx-zone boot
```

3 When the boot completes, use the `list` subcommand with the `-v` option to verify the status.

```
global# zoneadm list -v
```

You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>1</td>
<td>lx-zone</td>
<td>running</td>
<td>/export/home/lx-zone</td>
<td><code>lx</code></td>
<td>shared</td>
</tr>
</tbody>
</table>

**Example 33–5** Specifying Boot Arguments for Zones

Boot a zone using the `-i altinit` option:

```
global# zoneadm -z lx-zone boot -- -i /path/to/process
```

**Troubleshooting** If a message indicating that the system was unable to find the netmask to be used for the IP address specified in the zone’s configuration displays, see “netmasks Warning Displayed When Booting Zone” on page 406. Note that the message is only a warning and the command has succeeded.

▼ **How to Boot an \l x\ Branded Zone in Single-User Mode**

You must be the global administrator in the global zone to perform this procedure.

1 **Become superuser, or assume the Primary Administrator role.**

To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2 **Boot the zone in single-user mode.**

```
global# zoneadm -z lx-zone boot -- -s
```

**Where to Go From Here**

To log in to the zone, see configuration, see “Logging In to an \l x\ Branded Zone” on page 472.
Halting, Rebooting, Uninstalling, Cloning, and Deleting \( \text{lx} \) Branded Zones (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halt a zone.</td>
<td>The halt procedure is used to remove both the application environment and the virtual platform for a zone. The procedure returns a zone in the ready state to the installed state. To cleanly shut down a zone, see &quot;How to Use z\login to Shut Down an ( \text{lx} ) Branded Zone&quot; on page 476.</td>
<td>&quot;How to Halt an ( \text{lx} ) Branded Zone&quot; on page 463</td>
</tr>
<tr>
<td>Reboot a zone.</td>
<td>The reboot procedure halts the zone and then boots it again.</td>
<td>&quot;How to Reboot an ( \text{lx} ) Branded Zone&quot; on page 464</td>
</tr>
<tr>
<td>Uninstall a zone.</td>
<td>This procedure removes all of the files in the zone’s root file system. Use this procedure with caution. The action is irreversible.</td>
<td>&quot;How to Uninstall a Branded Zone&quot; on page 465</td>
</tr>
<tr>
<td>Provision a new non-global zone based on the configuration of an existing zone on the same system.</td>
<td>Cloning a zone is an alternate, faster method of installing a zone. You must still configure the new zone before you can install it.</td>
<td>&quot;Cloning an ( \text{lx} ) Branded Zone on the Same System&quot; on page 466</td>
</tr>
<tr>
<td>Delete a non-global zone from the system.</td>
<td>This procedure completely removes a zone from a system.</td>
<td>&quot;Deleting an ( \text{lx} ) Branded Zone From the System&quot; on page 469</td>
</tr>
</tbody>
</table>

Halting, Rebooting, and Uninstalling \( \text{lx} \) Branded Zones

**How to Halt an \( \text{lx} \) Branded Zone**

The halt procedure is used to remove both the application environment and the virtual platform for an \( \text{lx} \) branded zone. To cleanly shut down a zone, see How to Use z\login to Shut Down an \( \text{lx} \) Branded Zone.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**

To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in *System Administration Guide: Basic Administration.*
2 List the zones running on the system.
   global# zoneadm list -v
   
   You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>1</td>
<td>lx-zone</td>
<td>running</td>
<td>/export/home/</td>
<td>lx</td>
<td>shared</td>
</tr>
</tbody>
</table>

3 Use the zoneadm command with the -z option, the name of the zone, for example, lx-zone, and the halt subcommand to halt the given zone.
   global# zoneadm -z lx-zone halt

4 List the zones on the system again, to verify that lx-zone has been halted.
   global# zoneadm list -iv
   
   You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>lx-zone</td>
<td>installed</td>
<td>/export/home/</td>
<td>lx</td>
<td>shared</td>
</tr>
</tbody>
</table>

5 Boot the zone if you want to restart it.
   global# zoneadm -z lx-zone boot

Troubleshooting
If the zone does not halt properly, see “Zone Does Not Halt” on page 407 for troubleshooting tips.

▼ How to Reboot an lx Branded Zone
You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 List the zones running on the system.
   global# zoneadm list -v
   
   You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>1</td>
<td>lx-zone</td>
<td>running</td>
<td>/export/home/</td>
<td>lx</td>
<td>shared</td>
</tr>
</tbody>
</table>
3 **Use the** `zoneadm` **command with the** `-z reboot` **option to reboot the zone** `lx-zone`.
   
   ```
   global# zoneadm -z lx-zone reboot
   ```

4 **List the zones on the system again to verify that** `lx-zone` **has been rebooted.**
   
   ```
   global# zoneadm list -v
   ```

   You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>2</td>
<td>lx-zone</td>
<td>running</td>
<td>/export/home/lx-zone</td>
<td>lx</td>
<td>shared</td>
</tr>
</tbody>
</table>

   **Tip** – Note that the zone ID for `lx-zone` has changed. The zone ID generally changes after a reboot.

---

**How to Uninstall a Branded Zone**

**Caution** – This procedure removes all of the files in the zone’s root file system. The action is irreversible.

The zone cannot be in the running state. The `uninstall` operation is invalid for running zones.

You must be the global administrator in the global zone to perform this procedure.

1 **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in *System Administration Guide: Basic Administration.*

2 **List the zones on the system.**
   
   ```
   global# zoneadm list -v
   ```

   You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td></td>
<td>lx-zone</td>
<td>installed</td>
<td>/export/home/lx-zone</td>
<td>lx</td>
<td>shared</td>
</tr>
</tbody>
</table>

3 **Use the** `zoneadm` **command with the** `-z uninstall` **option to remove the zone** `lx-zone`.
   
   You can also use the `-F` option to force the action. If this option is not specified, the system will prompt for confirmation.
   
   ```
   global# zoneadm -z lx-zone uninstall -F
   ```
Note that when you uninstall a zone that has its own ZFS file system for the zonepath, the ZFS file system is destroyed.

4 List the zones on the system again, to verify that lx-zone is no longer listed.

   global# zoneadm list -v

You will see a display that is similar to the following:

   ID  NAME  STATUS  PATH  BRAND  IP
   0  global  running  /  native  shared

Troubleshooting If a zone uninstall is interrupted, the zone is left in the incomplete state. Use the zoneadm uninstall command to reset the zone to the configured state.

Use the uninstall command with caution because the action is irreversible.

Cloning an lx Branded Zone on the Same System

Cloning is used to provision a new zone on a system by copying the data from a source zonepath to a target zonepath.

When the source zonepath and the target zonepath both reside on ZFS and are in the same pool, the zoneadm clone command automatically uses ZFS to clone the zone. However, you can specify that the ZFS zonepath be copied and not ZFS cloned.

▼ How to Clone an lx Branded Zone

You must configure the new zone before you can install it. The parameter passed to the zoneadm create subcommand is the name of the zone to clone. This source zone must be halted.

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Halt the source zone to be cloned, which is lx-zone in this procedure.

   global# zoneadm -z lx-zone halt

3 Start configuring the new zone by exporting the configuration of the source zone lx-zone to a file, for example, master.

   global# zonercfg -z lx-zone export -f /export/zones/master
Note – You can also create the new zone configuration using the procedure “How to Configure the Zone” on page 249 instead of modifying an existing configuration. If you use this method, skip ahead to Step 6 after you create the zone.

4 Edit the file `master`. At a minimum, you must set a different zonepath and IP address for the new zone.

5 Create the new zone, `zone1`, by using the commands in the file `master`.
   global# zonecfg -z zone1 -f /export/zones/master

6 Install the new zone, `zone1`, by cloning `lx-zone`.
   global# zoneadm -z zone1 clone lx-zone

   The system displays:

   Cloning zonepath /export/home/lx-zone...

   If the source zonepath is on a ZFS pool, for example, `zeepool`, the system displays:

   Cloning snapshot zeepool/zones/lx-zone@SUNWzone1
   Instead of copying, a ZFS clone has been created for this zone.

7 List the zones on the system.
   global# zoneadm list -iv

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>lx-zone</td>
<td>installed</td>
<td>/export/home/lx-zone</td>
<td>lx</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>zone1</td>
<td>installed</td>
<td>/export/home/zone1</td>
<td>lx</td>
<td>shared</td>
</tr>
</tbody>
</table>

More Information

When a Source zonepath on a ZFS File System Is Cloned

When the `zoneadm` command clones a source zonepath that is on its own ZFS file system, the following actions are performed:

- The `zoneadm` command takes a software inventory.
- The `zoneadm` command takes a ZFS snapshot and names it `SUNWzoneX`, for example, `SUNWzone1`.
- The `zoneadm` command uses ZFS clone to clone the snapshot.

▼ How to Clone a Zone from an Existing Snapshot

You can clone a source zone multiple times from an existing snapshot that was originally taken when you cloned a zone.
You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Configure the zone** zone2.

3. **Specify that an existing snapshot be used to create** new-zone2.
   
   ```
   global# zoneadm -z zone2 clone -s zeepool/zones/lx-zone@SUNWzone1 lx-zone
   ```
   
   The system displays:

   Cloning snapshot zeepool/zones/lx-zone@SUNWzone1
   
   The `zoneadm` command validates the software from the snapshot `SUNWzone1`, and clones the snapshot.

4. **List the zones on the system.**
   
   ```
   global# zoneadm list -iv
   ```
   
<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>lx-zone</td>
<td>installed</td>
<td>/zeepool/zones/lx-zone</td>
<td>lx</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>zone1</td>
<td>installed</td>
<td>/zeepool/zones/zone1</td>
<td>lx</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>zone2</td>
<td>installed</td>
<td>/zeepool/zones/zone1</td>
<td>lx</td>
<td>shared</td>
</tr>
</tbody>
</table>

**▼ How to Use Copy Instead of ZFS Clone**

Use this procedure to prevent the automatic cloning of a zone on a ZFS file system by specifying that the zone path should be copied instead.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Specify that the zone path on ZFS be copied and not ZFS cloned.**
   
   ```
   global# zoneadm -z zone1 clone -m copy lx-zone
   ```
Deleting an \texttt{lx} Branded Zone From the System

The procedure described in this section completely deletes a zone from a system.

\section*{\textbf{How to Remove an \texttt{lx} Branded Zone}}

1 \hspace{1em} \textbf{Shut down the zone \texttt{lx-zone}.}
   \texttt{global\# zlogin lx-zone shutdown -y -g0 -i0}

2 \hspace{1em} \textbf{Remove the root file system for \texttt{lx-zone}.}
   \texttt{global\# zoneadm -z lx-zone uninstall -F}

3 \hspace{1em} \textbf{Delete the configuration for \texttt{lx-zone}.}
   \texttt{global\# zonecfg -z lx-zone delete -F}

4 \hspace{1em} \textbf{List the zones on the system, to verify that \texttt{lx-zone} is no longer listed.}
   \texttt{global\# zoneadm list -iv}

   You will see a display that is similar to the following:

   \begin{tabular}{lllll}
   ID & NAME & STATUS & PATH & BRAND & IP \\
   0 & global & running & / & native & shared \\
   \end{tabular}
Logging In to lx Branded Zones (Tasks)

This chapter provides the following information:

- Introductory information about zone login
- Completing the internal configuration of an installed lx branded zone
- Logging into the zone from the global zone
- Shutting down the zone
- Using the zonename command to print the name of the current zone

*zlogin* **Command Overview**

The *zlogin* command is used to log in from the global zone to any zone that is in the running state or the ready state.

**Note** – Only the *zlogin* command with the -C option can be used to log in to a zone that is not in the running state.

Unless the -C option is used to connect to the zone console, logging in to a zone using *zlogin* starts a new task. A task cannot span two zones.

As described in “How to Use Non-Interactive Mode to Access an lx Branded Zone” on page 474, you can use the *zlogin* command in non-interactive mode by supplying a command to run inside a zone. However, the command or any files the command acts upon cannot reside on NFS. The command will fail if any of its open files or any portion of its address space resides on NFS. The address space includes the command executable itself and the command’s linked libraries.

The *zlogin* command can only be used by the global administrator operating in the global zone. See the *zlogin(1)* man page for more information.
Branded Zone Login Methods

An overview of zone console and user login methods is provided in "Non-Global Zone Login Methods" on page 290.

The failsafe mode is used when a login problem occurs that prevents you from using the zlogin command or the zlogin command with the -C option to access the zone. This mode is described in "Failsafe Mode" on page 291.

Information on remote login zone is provided in "Remote Login" on page 291.

Interactive mode allocates a new pseudo-terminal for use inside the zone. Non-interactive mode is used to run shell-scripts which administer the zone. See "Interactive and Non-Interactive Modes" on page 292 for more information.

Login Procedures for Branded Zones (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log in to the zone.</td>
<td>You can log into a zone through the console, by using interactive mode to allocate a pseudo-terminal, or by supplying a command to be run in the zone. Supplying a command to be run does not allocate a pseudo-terminal. You can also log in by using failsafe mode when a connection to the zone is denied.</td>
<td>&quot;Logging In to an \l\x Branded Zone&quot; on page 472</td>
</tr>
<tr>
<td>Exit a branded zone.</td>
<td>Disconnect from a branded zone.</td>
<td>&quot;How to Exit the \l\x Branded Zone&quot; on page 475</td>
</tr>
<tr>
<td>Shut down a branded zone.</td>
<td>Shut down a branded zone by using the shutdown utility or a script.</td>
<td>&quot;How to Use zlogin to Shut Down an \l\x Branded Zone&quot; on page 476</td>
</tr>
</tbody>
</table>

Logging In to an \l\x Branded Zone

Use the zlogin command to log in from the global zone to any zone that is running or in the ready state. See the zlogin(1) man page for more information.

You can log in to a zone in various ways, as described in the following procedures. You can also log in remotely, as described in "Remote Login" on page 291.
How to Log In to the \texttt{lx} Branded Zone Console

You must be the global administrator in the global zone to perform this procedure.

1 \textbf{Become superuser, or assume the Primary Administrator role.}

To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 \textbf{Use the \texttt{zlogin} command with the -C option and the name of the zone, for example, \texttt{lx-zone}.}

\begin{verbatim}
 global# zlogin -C lx-zone
 [Connected to zone 'lx-zone' console]
\end{verbatim}

\textbf{Note} – If you start the \texttt{zlogin} session immediately after issuing the \texttt{zoneadm boot} command, boot messages from the zone will display:

\begin{verbatim}
INIT: version 2.85 booting
   Welcome to CentOS
      Press 'I' to enter interactive startup.
 Configuring kernel parameters: [ OK ]
 Setting hostname \texttt{lx-zone}: [ OK ]
[...]
 CentOS release 3.6 (Final)
 Kernel 2.4.21 on an i686
\end{verbatim}

3 \textbf{When the zone console displays, log in as root, press Return, and type the root password when prompted.}

\texttt{lx-zone console login: root}
Password:

\textbf{Note} – Recall that the root (superuser) password is root when the zone is installed from the Sun tarball. The root (superuser) password is unset (blank) when the zone is installed from ISO images or a CD.

How to Use Interactive Mode to Access a Branded Zone

In interactive mode, a new pseudo-terminal is allocated for use inside the zone.

You must be the global administrator in the global zone to perform this procedure.

1 \textbf{Become superuser, or assume the Primary Administrator role.}

To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.
2 From the global zone, log in to the zone, for example, \texttt{lx-zone}.
   
   \begin{verbatim}
   global# zlogin lx-zone
   
   Information similar to the following will display:
   [Connected to zone \texttt{lx-zone} pts/2]
   Last login: Wed Jul 3 16:25:00 on console
   Sun Microsystems Inc. SunOS 5.10 Generic July 2006
   
   exit
   \end{verbatim}

3 Type \texttt{exit} to close the connection.
   
   You will see a message similar to the following:
   [Connection to zone \texttt{lx-zone} pts/2 closed]

\section*{How to Verify the Running Environment}

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in \textit{System Administration Guide: Basic Administration}.

2 Log in to the zone, for example, \texttt{lx-zone}.
   
   \begin{verbatim}
   global# zlogin lx-zone
   
   \end{verbatim}

3 Verify that you are running in a Linux environment under the Solaris Operating System.
   
   \begin{verbatim}
   [root@lx-zone root]# `uname -a`
   
   You will see a display similar to the following:
   
   Linux lx-zone 2.4.21 BrandZ fake linux i686 i686 i386 GNU/Linux
   \end{verbatim}

\section*{How to Use Non-Interactive Mode to Access an \texttt{lx} Branded Zone}

Non-interactive mode is enabled when the user supplies a command to be run inside the zone. Non-interactive mode does not allocate a new pseudo-terminal.

Note that the command or any files that the command acts upon cannot reside on NFS.

You must be the global administrator in the global zone to perform this procedure.
1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 From the global zone, log in to the \texttt{lx-zone} zone and supply a command name.
   Replace command with the name of the command to be run inside the zone.
   \texttt{global\# zlogin lx-zone command}

\textbf{Example 34–1} Using the Command \texttt{uptime} in the Zone \texttt{lx_master}

\texttt{global\# zlogin lx_master uptime}
\begin{verbatim}
21:16:01 up 2:39, 0 users, load average: 0.19, 0.13, 0.11
fireball\#
\end{verbatim}

\textbf{▼ How to Exit the \texttt{lx} Branded Zone}

- To disconnect from a non-global zone, use the tilde (\texttt{~}) character and a period:
  \texttt{zonename\# ~.}

  Your screen will look similar to this:

  [Connection to zone 'lx-zone' pts/6 closed]

  \begin{itemize}
  \item You can also type \texttt{exit} to exit the zone.
  \end{itemize}

\textbf{See Also} For more information about \texttt{zlogin} command options, see the \texttt{zlogin(1)} man page.

\textbf{▼ How to Use Failsafe Mode to Enter an \texttt{lx} Branded Zone}

When a connection to the zone is denied, the \texttt{zlogin} command can be used with the \texttt{-S} option to enter a minimal environment in the zone.

You must be the global administrator in the global zone to perform this procedure.

1 Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

2 From the global zone, use the \texttt{zlogin} command with the \texttt{-S} option to access the zone, for example, \texttt{lx-zone}.
   \texttt{global\# zlogin -S lx-zone}
How to Use *zlogin* to Shut Down an *lx* Branded Zone

**Note** – Running `init 0` in the global zone to cleanly shut down a Solaris system also runs `init 0` in each of the non-global zones on the system. Note that `init 0` does not warn local and remote users to log off before the system is taken down.

Use this procedure to cleanly shut down a zone. To halt a zone without running shutdown scripts, see “How to Halt a Zone” on page 281.

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Log in to the zone to be shut down, for example, *lx-zone*, and specify *shutdown* as the name of the utility and *init 0* as the state.**
   
   `global# zlogin lx-zone shutdown -y -g0 -i 0`

   Your site might have its own shutdown script, tailored for your specific environment.

**More Information** – Using *shutdown* in Non-Interactive Mode

You cannot use the `shutdown` command in non-interactive mode to place the zone in single—user state at this time. See 6214427 for more information.

You can use an interactive login as described in “How to Use Interactive Mode to Access a Branded Zone” on page 473.
Moving and Migrating \texttt{lx} Branded Zones (Tasks)

This chapter describes how to:

- Move an existing \texttt{lx} branded zone to a new location on the same machine.
- Validate what will happen in an \texttt{lx} branded zone migration before the actual migration is performed.
- Migrate an existing \texttt{lx} branded zone to a new machine.

Moving an \texttt{lx} Branded Zone

This procedure is used to move a zone to a new location on the same system by changing the zonepath. The zone must be halted. The new zonepath must be on a local file system. The normal zonepath criteria described in “Resource and Property Types” on page 433 apply.

▼ How to Move a Zone

1. **Become superuser, or assume the Primary Administrator role.**
   Roles are described in “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. **Halt the zone to be moved, \texttt{db-zone} in this procedure.**
   ```
   global# zoneadm -z db-zone halt
   ```

3. **Use the \texttt{zoneadm} command with the \texttt{move} subcommand to move the zone to a new zonepath, \texttt{/export/zones/db-zone}.**
   ```
   global# zoneadm -z db-zone move /export/zones/db-zone
   ```
Verify the path.

```
$ zoneadm list -iv
```

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>lx-zone</td>
<td>installed</td>
<td>/export/home/lx-zone</td>
<td>lx</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>db-zone</td>
<td>installed</td>
<td>/export/zones/db-zone</td>
<td>lx</td>
<td>shared</td>
</tr>
</tbody>
</table>

**Migrating an \texttt{lx} Branded Zone to a Different Machine**

You can do a trial run of a zone migration before you actually move the zone to a different machine. For more information, see “About Validating a Zone Migration Before the Migration Is Performed” on page 311.

Note that the trial run does not validate the processor type, so you must verify that the target machine is running a supported processor.

**About Migrating an \texttt{lx} Branded Zone**

The `zonecfg` and `zoneadm` commands can be used to migrate an existing non-global zone from one system to another. The zone is halted and detached from its current host. The `zonestatus` is moved to the target host, where it is attached.

The following requirements apply to \texttt{lx} branded zone migration:

- The global zone on the target system must be running the same Solaris release as the original host.
- To ensure that the zone will run properly, the target system must have the same versions of the required operating system packages and patches that were installed on the original host.
- The brand must be the same on the original host and on the target system.
- The target system must have one of the following supported i686 processor types:
  - Intel
    - Pentium Pro
    - Pentium II
    - Pentium III
    - Celeron
    - Xeon
    - Pentium 4
    - Pentium M
    - Pentium D
    - Pentium Extreme Edition
    - Core
    - Core 2
The zoneadm detach process creates the information necessary to attach the zone on a different system. The zoneadm attach process verifies that the target machine has the correct configuration to host the zone. Because there are several ways to make the zonepath available on the new host, the actual movement of the zonepath from one system to another is a manual process that is performed by the global administrator.

When attached to the new system, the zone is in the installed state.

▼ How to Migrate an \textit{lx} Branded Zone

1 \textbf{Become superuser, or assume the Primary Administrator role.}  
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in \textit{System Administration Guide: Basic Administration}.

2 \textbf{Halt the zone to be migrated, \textit{lx}-zone in this procedure.}  
   \texttt{host1\# zoneadm -z lx-zone halt}

3 \textbf{Detach the zone.}  
   \texttt{host1\# zoneadm -z lx-zone detach}
   The detached zone is now in the configured state.

4 \textbf{Move the zonepath for \textit{lx}-zone to the new host.}  
   See “How to Move the zonepath to a new Host” on page 481 for more information.

5 \textbf{On the new host, configure the zone.}  
   \texttt{host2\# zonecfg -z lx-zone}
   You will see the following system message:

   \texttt{lx-zone: No such zone configured}
   Use ‘create’ to begin configuring a new zone.
6 To create the zone `lx-zone` on the new host, use the `zonecfg` command with the `-a` option and the `zonepath` on the new host.

```
zonecfg:lx-zone> create -a /export/zones/lx-zone
```

7 View the configuration.

```
zonecfg:lx-zone> info
zonename: lx-zone
zonepath: /export/zones/lx-zone
brand: lx
autoboot: false
bootargs:
pool: 
limitpriv: 
net:
  address: 192.168.0.90
  physical: bge0
```

8 (Optional) Make any required adjustments to the configuration.

For example, the network physical device might be different on the new host, or devices that are part of the configuration might have different names on the new host.

```
zonecfg:lx-zone> select net physical=bge0
zonecfg:lx-zone:net> set physical=e1000g0
zonecfg:lx-zone:net> end
```

9 Commit the configuration and exit.

```
zonecfg:lx-zone> commit
zonecfg:lx-zone> exit
```

10 Attach the zone on the new host.

- Attach the zone with a validation check.

  ```
  host2# zoneadm -z lx-zone attach
  ```

  The system administrator is notified of required actions to be taken if either or both of the following conditions are present:

  - Required packages and patches are not present on the new machine.
  - The software levels are different between machines.

- Force the attach operation without performing the validation.

  ```
  host2# zoneadm -z lx-zone attach -F
  ```
Caution – The -F option allows you to force the attach with no validation performed. This is useful in certain cases, such as in a clustered environment or for backup and restore operations, but it does require that the system be properly configured to host the zone. An incorrect configuration could result in undefined behavior later.

How to Move the zonopath to a new Host

There are many ways to create an archive of the zonopath. For example, you can use the cpio or pax commands described in the cpio(1) and pax(1) man pages.

There are also several ways to transfer the archive to the new host. The mechanism used to transfer the zonopath from the source host to the destination depends on the local configuration. In some cases, such as a SAN, the zonopath data might not actually move. The SAN might simply be reconfigured so the zonopath is visible on the new host. In other cases, the zonopath might be written to tape, and the tape mailed to a new site.

For these reasons, this step is not automated. The system administrator must choose the most appropriate technique to move the zonopath to the new host.

1. Become superuser, or assume the Primary Administrator role.
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2. Move the zonopath to the new host. You can use the method described in this procedure, or use another method of your choice.

Example 35–1 Archiving and Moving the zonopath Using the tar Command

1. Create a tar file of the zonopath on host1 and transfer it to host2 by using the sftp command.

```
host1# cd /export/zones
host1# tar cf lx-zone.tar lx-zone
host1# sftp host2
Connecting to host2...
Password: 
sftp> cd /export/zones
sftp> put lx-zone.tar
Uploading lx-zone.tar to /export/zones/lx-zone.tar
sftp> quit
```

2. On host2, unpack the tar file.

```
host2# cd /export/zones
host2# tar xf lx-zone.tar
```
For more information, see `sftp(1)` and `tar(1)`.

**Troubleshooting**

See “Resolving Problems With a `zoneadm attach` Operation” on page 407 for troubleshooting information on the following:

- Patches and packages are out of sync.
- Operating system releases do not match.

The user must verify that the processor type in the new machine is supported. See "About Migrating an \(\text{l}x\) Branded Zone" on page 478 for more information.

### About Validating an \(\text{l}x\) Branded Zone Migration Before the Migration Is Performed

You can perform a trial run before the zone is moved to the new machine by using the “no execute” option, `-n`.

The `zoneadm detach` subcommand is used with the `-n` option to generate a manifest on a running zone without actually detaching the zone. The state of the zone on the originating system is not changed. The zone manifest is sent to `stdout`. The global administrator can direct this output to a file or pipe it to a remote command to be immediately validated on the target host. The `zoneadm attach` subcommand is used with the `-n` option to read this manifest and verify that the target machine has the correct configuration to host the zone without actually doing an attach.

The zone on the target system does **not** have to be configured on the new host before doing a trial-run attach.

### How to Validate an \(\text{l}x\) Branded Zone Migration Before the Migration Is Performed

You must be the global administrator in the global zone to perform this procedure.

1. **Become superuser, or assume the Primary Administrator role.**
   
   To create the role and assign the role to a user, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Use one of the following methods.**

   - Generate the manifest on a source host named \(\text{l}x\)-zone and pipe the output to a remote command that will immediately validate the target host:
     
     ```
     global# zoneadm -z \(\text{l}x\)-zone detach -n | ssh remotehost zoneadm attach -n -
     ```
The hyphen (−) at the end of the line specifies stdin for the path.

- Generate the manifest on a source host named lx-zone and direct the output to a file:
  global# zoneadm -z lx-zone detach -n

  Copy the manifest to the new host system as described in “How to Move the zonepath to a new Host” on page 481, and perform the validation:

  global# zoneadm attach -n path_to_manifest

  The path can be − to specify stdin.
Administering and Running Applications in \texttt{lx} Branded Zones (Tasks)

This chapter contains material on running applications in an \texttt{lx} branded zone.

**About Maintaining a Supported Configuration**

When you installed a zone with a supported CentOS or Red Hat Enterprise Linux distribution, you created a supported zone. If you add packages from different versions to this zone, it is possible to create a branded zone that cannot be supported.

**Upgrading the Distribution and Adding Packages**

**▼ How to Upgrade a CentOS 3.x Distribution**

You must be the zone administrator in the \texttt{lx} branded zone to perform this procedure.

- **Upgrade a CentOS 3.x distribution to a different version by using** \texttt{yum upgrade} or \texttt{up2date}.
  
  For instructions, see the documentation available at \url{http://www.centos.org}.

**▼ How to Upgrade a Red Hat 3.x Distribution**

You must be the zone administrator in the \texttt{lx} branded zone to perform this procedure.

- **Update a Red Hat Enterprise Linux 3.x distribution to a different version by using** \texttt{up2date}.
  
  For instructions, see the documentation available at \url{http://www.redhat.com}. 
How to Upgrade a Package

You must be the zone administrator in the \( \text{lx} \) branded zone to perform this procedure.

To update a package, use one of the following methods.

- `yum update package_name`
- `rpm -U package_name`

More Information

Using `yum` and `rpm`

`yum`:

- The document *Software Management with Yum* includes a chapter on installing software from an isolated package. See [http://fedora.redhat.com/docs/yum](http://fedora.redhat.com/docs/yum).
- `yum.conf(5)`
- `yum(8)`

`rpm`:

- `rpm(8)`

How to Install an Application in an \( \text{lx} \) Branded Zone

Applications are installed as they are on a Linux system, by mounting the CD and running the installation program. This section covers a typical application installation in an \( \text{lx} \) branded zone.

**Tip** – If you know you will be using CDs or DVDs to install applications in an \( \text{lx} \) branded zone, add read-only access to CD or DVD media in the global zone when you initially configure the branded zone. See step 7 in “How to Install MATLAB 7.2 Using CDs” on page 487.

About MATLAB

MATLAB is a high-level language and interactive environment that enables you to perform computationally intensive tasks quickly. The product was developed by The MathWorks. See [http://www.mathworks.com](http://www.mathworks.com) for more information.
How to Install MATLAB 7.2 Using CDs

1 Obtain the MATLAB 7.2 CDs.
   There are three CDs in the MATLAB/Simulink package. Only discs 1 and 3 are needed for a simple MATLAB installation.

2 Create and install an \texttt{l}x branded zone as described in "How to Configure, Verify, and Commit the \texttt{l}x Branded Zone" on page 441 and "Installing and Booting \texttt{l}x Branded Zones" on page 456.

3 If the Volume Management file system is not running in the global zone, start it.
   
   \texttt{global\# svcadm volfs enable}

4 Insert the media.

5 Check for media in the drive.
   
   \texttt{global\# volcheck}

6 Test whether the CD is automounted.
   
   \texttt{global\# ls /cdrom}
   
   You will see a display similar to the following:

   \begin{verbatim}
   cdrom   cdrom1   mathworks_2006a1
   \end{verbatim}

7 Loopback mount the file system with the options \texttt{ro,nodevices} (read-only and no devices) in the non-global zone.
   
   \texttt{global\# zonecfg -z \texttt{l}x-zone}
   \texttt{zonecfg:\texttt{l}x-zone> add fs}
   \texttt{zonecfg:\texttt{l}x-zone:fs> set dir=/cdrom}
   \texttt{zonecfg:\texttt{l}x-zone:fs> set special=/cdrom}
   \texttt{zonecfg:\texttt{l}x-zone:fs> set type=lofs}
   \texttt{zonecfg:\texttt{l}x-zone:fs> add options [ro,nodevices]}
   \texttt{zonecfg:\texttt{l}x-zone:fs> end}
   \texttt{zonecfg:\texttt{l}x-zone> commit}
   \texttt{zonecfg:\texttt{l}x-zone> exit}

8 Reboot the non-global zone.
   
   \texttt{global\# zoneadm -z \texttt{l}x-zone reboot}

9 Use the \texttt{zoneadm list} command with the \texttt{-v} option to verify the status.
   
   \texttt{global\# zoneadm list -v}
How to Install an Application in an lx Branded Zone

You will see a display that is similar to the following:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>1</td>
<td>lx-zone</td>
<td>running</td>
<td>/export/home/lx-zone</td>
<td>lx</td>
<td>shared</td>
</tr>
</tbody>
</table>

10 Log in to the lx zone.
global# zlogin lx-zone

11 Verify the CD-ROM mount.
lx-zone# ls /cdrom

You will see a display similar to this:

cdrom cdrom1 mathworks_2006a1

12 Create the license file as described in the MATLAB documentation.

13 Install the product as described in the product installation guide.
lx-zone# /mnt/install

14 Exit the zone.
lx-zone# exit

Tip — You might want to retain the /cdrom file system in your non-global zone. The mount will always reflect the current contents of the CD-ROM drive, or an empty directory if the drive is empty.

15 (Optional) If you want to remove the /cdrom file system from the non-global zone, use the following procedure.
global# zonecfg -z lx-zone
zonecfg:lx-zone> remove fs dir=/cdrom
zonecfg:lx-zone> commit
zonecfg:lx-zone> exit

How to Install MATLAB 7.2 Using ISO Images

Before You Begin
Note that this method consumes considerable disk space.

1 Obtain the MATLAB 7.2 CDs.
There are three CDs in the MATLAB/Simulink package. Only discs 1 and 3 are needed for a simple MATLAB installation.
2 Create and install an \texttt{lx} branded zone as described in "How to Configure, Verify, and Commit the \texttt{lx} Branded Zone" on page 441 and "Installing and Booting \texttt{lx} Branded Zones" on page 456.

3 Copy the data from each CD to a \texttt{.iso} file.
   
   \texttt{global# /usr/bin/dd if=/dev/rdsk/c1d0s2 of=\texttt{disk1.iso}}
   
   This copies the data from the first CD to the file \texttt{disk1.iso}. Repeat, using a different file name such as \texttt{disk3.iso}, for the third CD.

4 From the global zone, \texttt{lofi}-mount the first \texttt{.iso} file in the \texttt{lx} zone.
   
   \texttt{global# lofiadm -a /zpool/local/disk1.iso}
   \texttt{global# mount -F hsfs /dev/lofi/1 /zones/lx-zone/root/mnt}

5 Log in to the \texttt{lx} zone.
   
   \texttt{global# zlogin lx-zone}

6 Use X forwarding to redirect the display to your desktop:
   
   \texttt{lx-zone# ssh -X root@lx-zone}

7 Create the license file as described in the MATLAB documentation.

8 Install the product as described in the product installation guide.
   
   \texttt{lx-zone# /mnt/install}

9 When prompted to insert CD 3, go back to the global zone terminal window and mount \texttt{disk3.iso} file in place of the first.
   
   \texttt{global# umount /zones/lx-zone/root/mnt}
   \texttt{global# lofiadm -d /dev/lofi/1}
   \texttt{global# lofiadm -a /zpool/local/disk3.iso}
   \texttt{global# mount -F hsfs /dev/lofi/1 /zones/lx-zone/root/mnt}

The installation will finish.

### Backing Up \texttt{lx} Branded Zones

For information on zone backup, see "About Backing Up a Solaris System With Zones Installed" on page 374, "Determining What to Back Up in Non-Global Zones" on page 375, "Backing Up a Solaris System With Installed Zones" on page 399, "About Restoring Non-Global Zones" on page 376, and "Restoring a Non-Global Zone" on page 402.
The exclusive-IP network configuration is not supported in an \l branded zone. Only the shared-IP network configuration is supported.

The chroot command is not supported in a Linux zone. If used on a process, that process would no longer be able to see the Solaris libraries it needs to run.

Although you can configure and install \l branded zones on a Solaris Trusted Extensions system that has labels enabled, you cannot boot \l branded zones on this system configuration.

You cannot add local Linux file systems using the fs resource property of the zonecfg command.
Sun xVM Hypervisor

The Sun™ xVM hypervisor within the OpenSolaris™ Operating System runs on x64 and x86 machines. The hypervisor is used to execute virtual machine instances called domains. The hypervisor is based on the work of the open source Xen project.
This chapter introduces the Sun™ xVM hypervisor. The hypervisor is derived from an open source project, Xen.

For more information about using the hypervisor and the xVM architecture, see:

- The OpenSolaris community site (http://www.opensolaris.org/os/community/xen), which provides the latest Sun xVM hypervisor documentation, including xVM 3.3 release documentation
- The home of the Xen project (http://www.xen.org)

Sun xVM Hypervisor Virtualization System Overview

The Sun xVM hypervisor is a type 1 hypervisor that partitions a single physical machine into multiple virtual machines, to provide server consolidation and utility computing. Existing applications and binaries run unmodified.

The hypervisor presents a virtual machine to guests. The hypervisor forms a layer between the software running in the virtual machine and the hardware. This separation enables the hypervisor to control how guest operating systems running inside a virtual machine use hardware resources.

The hypervisor securely executes multiple virtual machines, or guest domains, simultaneously on a single x64 or x86 compatible computer. Unlike virtualization using zones, each virtual machine runs a full instance of an operating system.

There are two kinds of domains, the control domain and the guest domain. The control domain is also known as domain 0, or dom0. A guest operating system, or unprivileged domain, is also called a domain U or domU.

When working with the xVM software, note that the virsh and virt-install commands are preferred over the use of the legacy xm command whenever possible.
Uniform View of Hardware

A hypervisor provides a uniform view of underlying hardware. Machines from different vendors with different I/O subsystems appear to be the same machine, which means that virtual machines can run on any available supported computer. Thus, administrators can view hardware as a pool of resources that can run arbitrary services on demand. Because the hypervisor also encapsulates a virtual machine’s software state, the hypervisor layer can map and remap virtual machines to available hardware resources at any time and also use live migration to move virtual machines across computers. These capabilities can also be used for load balancing among a collection of machines, dealing with hardware failures, and scaling systems. When a computer fails and must go offline or when a new machine comes online, the hypervisor layer can remap virtual machines accordingly. Virtual machines are also easy to replicate, which allows administrators to bring new services online as needed.

The hypervisor virtualizes the system’s hardware. A virtualization API and tools are provided by the libvirt and virt-install utilities. The hypervisor transparently shares and partitions the system’s CPUs, memory, and NIC resources among the user domains. The hypervisor performs the low-level work required to provide a virtualized platform for operating systems.

The hypervisor assigns one or more virtual CPUs (VCPPUs) to each domain, allocated from dom0. The virsh setvcpus and virsh vcpupin commands can be used to dynamically set and pin VCPPUs to processors. Each VCPU contains all the state one would typically associate with a physical CPU, such as registers, flags, and timestamps. A VCPU in xVM is an entity that can be scheduled, like a thread in the Solaris® system. When it is a domain’s turn to run on a CPU, xVM loads the physical CPU with the state in the VCPU, and lets it run. The Solaris system treats each VCPU as it would treat a physical CPU. When the hypervisor selects a VCPU to run, it will be running the thread that the Solaris system loaded on the VCPU.

When to Use Domains

Containment

Containment gives administrators a general-purpose undo capability. Administrators can suspend a virtual machine and resume it at any time, or checkpoint a virtual machine and roll it back to a previous execution state. With this capability, systems can more easily recover from crashes or configuration errors. See “Recovery” on page 544.

Containment also supports a very flexible mobility model. Users can copy a suspended virtual machine over a network or store and transport it on removable media. The hypervisor provides total mediation of all interactions between the virtual machine and underlying hardware, thus allowing strong isolation between virtual machines and supporting the multiplexing of many virtual machines on a single hardware platform. The hypervisor can consolidate several physical machines with low rates of utilization as virtual systems on a single computer, thereby lowering hardware costs and space requirements.
Security
Strong isolation is also valuable for reliability and security. Applications that previously ran together on one machine can now be separated on different virtual machines. If one application experiences a fault, the other applications are isolated from this occurrence and will not be affected. Further, if a virtual machine is compromised, the incident is contained to only that compromised virtual machine.

Resource Virtualization to Enable Interoperability
The hypervisor provides a layer between software environments and physical hardware that has the following characteristics:
- Programmable and transparent to the software above it
- Makes efficient use of the hardware below it

Virtualization provides a way to bypass interoperability constraints. Virtualizing a system or component such as a processor, memory, or an I/O device at a given abstraction level maps its interface and visible resources onto the interface and resources of an underlying, possibly different, real system. Consequently, the real system appears as a different virtual system or even as multiple virtual systems.

Hardware Platform for Running Sun xVM Hypervisor
Supported hardware includes x64 and x86 machines.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Description</th>
</tr>
</thead>
</table>
| Processors   | AMD or Intel x86/x64 (AMD-V or Intel-VT for HVM domains)  
Requires at least 2 cores, 1 for the control domain and at least 1 for the guest. |
| Memory       | Default configuration requires 4 GB. |
| I/O (Disk on which control domain 0 is installed) |  
- SCSI, iSCSI, Serial ATA (SATA) or Serial Attached SCSI (SAS).  
- Fibre channel to a JBOD (term for just a bunch of disks or drives).  
Note – Legacy ATA (IDE) drives are acceptable for devices such as CD-ROM. These drives are not recommended as the root or image drive on which control domain 0 is installed, for performance reasons. |
### Determining HVM Support

To run Windows hardware-assisted virtual machine (HVM) domains, an HVM-capable machine that is running a Solaris xVM dom0 is required. A machine is HVM-capable if it is has either an AMD Opteron Revision F or later, or an Intel CPU with VT extensions.

To determine whether a machine is HVM-capable, run the `virt-install` program with the `-v` option.

If HVM is not supported, an error message displays:

```
# virt-install -v
ERROR Host does not support virtualization type ‘hvm’
```

### Sun xVM Hypervisor Memory Requirements

The current 4–GB default configuration includes memory for the OpenSolaris xVM hypervisor control functions and 10 or fewer guests.

When dom0 is booted, the OpenSolaris xVM hypervisor reserves a small amount of memory for itself and assigns the remainder to dom0. When a domU is created, the Sun xVM hypervisor takes memory from dom0 and assigns it to the new domU.

There is no default configuration or a default amount of memory to assign to dom0, unless the user sets a `dom0_mem` value in the GRUB menu when booting dom0. The values that can be set are initial memory, minimum memory, and maximum memory.

If you are running known memory hungry apps in dom0, then you need more memory. Examples include cacao processes and Firefox.

### The Solaris System and x86 Platforms

With the introduction of the Sun xVM hypervisor, there are now two platform implementations on the x86 architecture. The implementation names are returned by the `uname` command with the `-i` option and refer to the same machine, 186pc.

- 186pc refers to the Solaris system directly installed on the physical system, also known as running on `bare metal`.
- 186xpv refers to the Solaris system running both paravirtualized and fully virtualized on top of an xVM hypervisor.
Applications should use the `uname` command with the `-p` option. On x86 platforms, regardless of whether the Solaris system is running under xVM, this option always returns i386. The `-p` option is equivalent to using the `-m` option in some other operating systems.

## Guests That Are Known to Work

Supported virtual machine configurations in the OpenSolaris 2009.06 release include the OpenSolaris control domain (domain 0), and guest, or domU, operating systems.

Because the control domain must work closely with the hypervisor layer, the control domain is always paravirtualized. A system can have both paravirtualized and fully virtualized domains running simultaneously.

Types of guests include the following:

- Fully virtualized guests, referred to as hardware-assisted virtual machines (HVMs). These are virtual machines that utilize Intel-VT and AMD-V extensions. An HVM guest domain runs an unmodified operating system.
- HVM + PVIO: These guests provide better performance through the use of PVIO drivers.
- Paravirtualized (PV): A PV guest OS is a modified OS that is aware that is operating under a hypervisor, which results in higher performance.

<table>
<thead>
<tr>
<th>Guest OS</th>
<th>Type of Guest</th>
<th>32 or 64-Bit</th>
<th>Uniprocessor or Multiprocessor Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 2003 SP2</td>
<td>HVM + PVIO</td>
<td>32-bit</td>
<td>MP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Obtain the Sun xVM Guest Additions (Early Access 3) drivers and Sun xVM Guest Additions (Early Access 3) Installation guide <a href="#">here</a>, and install in the guest. Reboot the guest from inside the guest.</td>
</tr>
<tr>
<td>Windows XP</td>
<td>HVM + PVIO</td>
<td>32-bit</td>
<td>MP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Obtain the OpenSolaris™ xVM Guest Additions (Early Access 3) drivers and Sun xVM Guest Additions (Early Access 3) Installation guide <a href="#">here</a>, and install in the guest. Reboot the guest from inside the guest.</td>
</tr>
<tr>
<td>Windows Server 2008</td>
<td>HVM + PVIO</td>
<td>32-bit</td>
<td>MP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Obtain the Sun xVM Guest Additions (Early Access 3) drivers and Sun xVM Guest Additions (Early Access 3) Installation guide <a href="#">here</a>, and install in the guest. Reboot the guest from inside the guest.</td>
</tr>
</tbody>
</table>
Guests That Are Known to Work

<table>
<thead>
<tr>
<th>Guest OS</th>
<th>Type of Guest</th>
<th>32 or 64–Bit</th>
<th>Unprocessor or Multiprocessor</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris 10 5/09 (S10U7) + PVIO</td>
<td>HVM + PVIO</td>
<td>64-bit</td>
<td>UP</td>
<td>The Solaris 10 5/09 (Solaris 10 Update 7) release is shipped with the Solaris PV drivers. A Solaris guest domain works like a normal Solaris Operating System. All of the expected tools are available.</td>
</tr>
<tr>
<td>Solaris 10 10/08 (S10U6) + PVIO</td>
<td>HVM + PVIO</td>
<td>64-bit</td>
<td>UP</td>
<td>The Solaris 10 10/08 (Solaris 10 Update 6) release is shipped with the Solaris PV drivers.</td>
</tr>
<tr>
<td>Solaris 10 5/08 (S10U5) + PVIO</td>
<td>HVM + PVIO</td>
<td>64-bit</td>
<td>UP</td>
<td>To run the Solaris 10 5/08 release as a guest, download Solaris 10 patch 137112-06 (or later) from SunSolve (<a href="http://sunsolve.sun.com">http://sunsolve.sun.com</a>) to obtain the Solaris PV drivers. The SunSolve site provides download instructions. After the patch is applied to the domain, perform the following steps: 1. Run <code>sys-unconfig(1M)</code>. 2. Verify that the file <code>/etc/driver_aliases</code> contains the line <code>xpv pci5853,1.1</code>. 3. Verify that the file <code>/etc/name_to_major</code> contains the following lines:   - <code>xpv 249</code>   - <code>xpvd 250</code>   - <code>xnf 251</code>   - <code>xdf 252</code> 4. Reboot the guest. Upon reboot, select the PV network interface (<code>xnf</code>).</td>
</tr>
<tr>
<td>Solaris Express Community Edition (SXCE) Build 111 or later</td>
<td>HVM + PVIO</td>
<td>64-bit</td>
<td>UP</td>
<td>SXCE 110b and later builds are shipped with the Solaris PV drivers. A Solaris guest domain works like a normal Solaris Operating System. All of the expected tools are available.</td>
</tr>
</tbody>
</table>
OpenSolaris 2008.11 and 2009.06
HVM + PVIO
PV
64-bit UP
OpenSolaris is shipped with the Solaris PV drivers.
Continue to update your system for the latest bug fixes and features.
A Solaris guest domain works like a normal Solaris Operating System. All of the expected tools are available.
For PV installation instructions, see “How to Install Open Solaris 2008.11 or later in Paravirtualized Mode,” below.

RHEL 5.3
HVM
64-bit UP

Caution – Note that Windows HVM domains can be susceptible to viruses, so make sure you comply with your site’s network security policies.

The following information applies to the control domain:
- ISA floppy is not supported.
- For 32-bit, the processor must support physical address extensions (PAE) mode.
- Beginning with build snv_83, the OpenSolaris OS supports all Ethernet-type interfaces, and their data-links can be administered with the `dladm` command. Prior to OpenSolaris build snv_83, the network interface card (NIC) must support GLDv3. These devices include `bge`, `e1000g`, `xge`, `nge`, and `rge`.

The Sun xVM Hypervisor and Domain 0

The hypervisor is responsible for controlling and executing each of the domains and runs with full privileges. The control tools for managing the domains run under the specialized control domain domain 0 (dom0).

The hypervisor virtualizes the system’s hardware. The hypervisor transparently shares and partitions the system’s CPUs, memory, and NIC resources among the user domains. The hypervisor performs the low-level work required to provide a virtualized platform for operating systems.

The hypervisor relies primarily on the control domain for the following:
- Which guest domains are created.
The Sun xVM Hypervisor and Domain 0

- Which resources the guest domains can access.
- How much memory a given guest domain can have.
- Which devices a guest domain can access, because xVM does not include any device drivers. The hypervisor delegates control of the machine’s physical devices to the control domain, domain 0.

Thus, by default, only the control domain has access to physical devices. The guest domains running on the host are presented with virtualized devices. The domain interacts with the virtualized devices in the same way that the domain would interact with the physical devices. Also see “Resource Virtualization to Enable Interoperability” on page 495.

The following figure shows the Sun xVM hypervisor configuration.

![Sun xVM Configuration Diagram](image-url)
Sun xVM Hypervisor Scheduler

The hypervisor schedules running domains (including domain 0) onto the set of physical CPUs as configured. The scheduler is constrained by configuration specifications such as the following.

- Domain configuration, such as the number of virtual CPUs allocated.
- Runtime configuration, such as pinning virtual CPUs to physical CPUs. Pinning CPUs ensures that certain VCPUs can only run on certain physical CPUs.

The default domain scheduler for the hypervisor is the credit scheduler. This is a fair-share domain scheduler that balances virtual CPUs of domains across the allowed set of physical CPUs according to workload. No CPU will be idle if a domain has work to do and wants to run.

The scheduler is configured through the `xm sched-credit` command described in the `xm(1M)` man page.

The following parameters are used to configure the scheduler:

- `-d domain`, `--domain=domain` Domain for which to set scheduling parameters.
- `-c cap`, `--cap=cap` The maximum amount of CPU the domain can consume. A value of zero, which is the default, means no maximum is set. The value is expressed in percentage points of a physical CPU. A value of 100 percent corresponds to one full CPU. Thus, a value of 50 specifies a cap of half a physical CPU.
- `-w weight`, `--weight=weight` The relative weight, or importance, of the domain. A domain with twice the weight of the other domains receives double the CPU time of those other domains when CPU use is in contention. Valid weights are in the range 1-65536. The default weight is 256.

**EXAMPLE 37-1  `xm sched-credit` Configuration**

The following line configures scheduling parameters for the domain `sol1`. The domain has a weight of 500 and a cap of 1 CPU.

```
xm sched-credit -d sol1 -w 500 -c100
```

If used without the `-w` and `-c` options, the current settings for the given domain are shown.
Supported Virtualization Modes

There are two basic types of virtualization, full virtualization and paravirtualization. The hypervisor supports both modes.

Full virtualization allows any x86 operating system, including Solaris, Linux, or Windows systems, to run in a guest domain.

Paravirtualization requires changes to the operating system, with the minimum requirement being modifications to support the virtual device interfaces.

A system can have both paravirtualized and fully virtualized domains running simultaneously.

For paravirtualized mode and for all types of operating systems, the only requirement is that the operating system be modified to support the virtual device interfaces.

Overview of Paravirtualization

In the more lightweight paravirtualization, the operating system is both aware of the virtualization layer and modified to support it, which results in higher performance.

The paravirtualized guest domain operating system is ported to run on top of the hypervisor, and uses virtual network, disk, and console devices.

Since the control domain must work closely with the hypervisor layer, control domain is always paravirtualized. Guest domains can be either paravirtualized or fully virtualized.

Devices and Drivers

Because full paravirtualization requires changes to the OS, only specific operating systems can be hosted in a fully paravirtualized guest domain. Currently, those are limited to Solaris, Linux FreeBSD, and NetBSD, although others might be made available in the future.

Partial paravirtualization describes a mechanism in which an otherwise unmodified OS is augmented with paravirtualized drivers for I/O devices. This can significantly improve the performance of the otherwise unmodified guest domain.

Paravirtualized drivers for I/O devices are implemented as a pair of drivers, one in each of the guest and host domains. This mechanism is often termed split drivers.

A frontend driver runs in the guest domain and communicates with a backend driver running in domain 0. This enables the guest domain to access services available to domain 0.
xVM software in the OpenSolaris release currently supports two main split drivers, one for network I/O and one for disk I/O.

Within the guest domain, the frontend driver appears as a normal device. For network, this is an Ethernet device. For disk, this is a traditional block device.

Within domain 0, the behavior of the backend driver depends on the type of device and the configuration of the domain. Network backend drivers typically extend the physical network connectivity available to domain 0 into the guest domain by using a virtual NIC feature. Disk backend drivers can make disk space available to guest domains by using files, ZFS volumes, and physical devices. Various file formats are supported when files are used to provide storage, for example, VMDK. (For more information on VMDK, see the section “Using vdiskadm to Create Virtual Disks” on page 506.)

The Solaris frontend drivers share the same source code whether they are used in paravirtualized (PV) or partially paravirtualized (HVM+PV) domains. There are #ifdefs in the driver source code to accommodate differences between HVM+PV and PV environments, but otherwise they are the same.

The Windows frontend drivers have different source code than those for Solaris, but the protocol between them and the Solaris backend drivers is the same as that used by the Solaris frontend drivers. This protocol was developed by the Xen open source community and is defined by the source code for the Linux variants of the drivers.

The source code for the Solaris drivers is found in /usr/src/uts/common/xen/io. The network frontend driver is xnf, and the disk frontend driver is xdf. The backend drivers have various names, such as xnb, xnbo, xnbu, xdb, and xpvtap.

In addition to these drivers, the Solaris console is virtualized when the Solaris system is running as a guest domain. The console driver interacts with the xenconsoled(1M) daemon running in domain 0 to provide console access.

**Overview of Full Virtualization**

In a full virtualization, the operating system is not aware that it is running in a virtualized environment under xVM. A fully virtualized guest domain is referred to as a hardware-assisted virtual machine (HVM). An HVM guest domain runs an unmodified operating system.

Fully-virtualized guest domains are supported under xVM with virtualization extensions available on Intel-VT or AMD Secure Virtual Machine (SVM) processors. These extensions must be present and enabled. Some BIOS versions disable the extensions by default. Note that this that hardware is also needed in HVM+PVIO configurations such as Solaris 10 5/09 (Solaris 10 U7) guest domains.
Virtual Devices

**Networks**

The physical network consists of both an external physical LAN and the extension of the LAN within the host to a guest’s network. Paravirtualized domains use the xmb backend driver in dom0 to communicate with a physical network interface.

The virtual network operates through the underlying physical network infrastructure.

You can create one physical network, also referred to as a switch, for each NIC on the system. If you have more than one physical NIC configured, you might want to configure the `default-nic` property of the xvm/xend SMF service, as described in the `xend(1M)`.

To view the IP address assigned through DHCP, use the `ifconfig` command.

See the document *New Network Options, Including Limiting Bandwidth and Setting a VLAN ID, for Virtual Network Interfaces Attached to a Guest* to learn about conventions for network options used in the `virt-install` utility.

**Virtual NICs**

A single physical NIC can be carved into multiple VNICS, which can be assigned to different Solaris xVM instances running on the same system. VNICS are managed using the `dladm` command line utility described in the `dladm(1M)` man page. You can use `virsh dumpxml` output to correlate the domain’s network interface with the assigned VNIC.

The `dladm show-vnic` command can be used to display VNIC information. In the following output, 1 is the domain ID, and 0 is the NIC number for that guest.

```
# dladm show-vnic
  LINK  OVER  SPEED  MACADDRESS  MACADRTYPE  VID
xvm1_0  e1000g0  1000  0:16:3e:06:99:4d  fixed  0
```
For more information, see OpenSolaris Project: Crossbow: Network Virtualization and Resource Control (http://www.opensolaris.org/os/project/crossbow). Crossbow provides network virtualization and resource control by virtualizing the stack and NIC around any service, protocol, or virtual machine.

**Virtual FibreChannel HBAs**

Certain FibreChannel HBAs that support N Port ID virtualization (NPIV) can be carved into multiple virtual HBAs, in a manner similar to VNICS. Each virtual HBA has a unique identity, or WWN, on the SAN, which can be used for LUN masking or zoning. Block devices present on these virtual HBAs can be automatically mounted to specific OpenSolaris xVM guest operating systems. When these guest operating systems migrate, the HBA’s identity also migrates. NPIV is administered using the `fcinfo(1M)` and `xm(1M)` commands.

**Virtual FibreChannel HBAs**

Virtual block devices can be stored on FibreChannel disk subject to the following limitations:

- Back end block device names (for example, `c0t0d0s2`) must be the same for migration to succeed. If device names are different, migration will fail.
- For FibreChannel devices, it is likely that the `cX` number will be different if `scsi_vhci`, described in `scsi_vhci(7D)`, is not used. If `scsi_vhci` is used, device names are likely to be identical. If `scsi_vhci` is enabled and has never been disabled, the controller should be at `cX` on x86/x64 machines.

**NPIV and FibreChannel**

NPIV is fully supported for xVM. The `xm` command can associate all block devices from one NPIV port to a guest domain. NPIV identity, specifically the port WWN and node WWN, will migrate between devices on the SAN. NPIV allows zoning and LUN masking to be used with Xen. Zoning and LUN masking are useful tools for access control on a SAN. Soft zoning rather than hard zoning (grouping by switch port) should be used on the switch. Soft zoning groups devices by the HBA’s Port WWN. If there is more than one physical SAN, or if the system uses hard zoning, the administrator must ensure that the physical HBA is connected to the correct SAN. Switch administrative tools can be used for this purpose.

**How to Configure NPIV for Hypervisor Use**

This procedure uses the `fcinfo` command described in `fcinfo(1M)`.

1. Identify the FibreChannel HBAs that will be used. If migration will be supported, the HBAs must be identified. The `fcinfo` command can be used to list the Port WWN.
2. Create NPIV ports in the Dom0 control domain.
3. View devices by using fcinfo in dom0. Verify that they are visible in the respective guest domains.

**Using vdiskadm to Create Virtual Disks**

You can use the `virt-install` command to create disks.

The `vdiskadm` command described in the `vdiskadm(1M)` man page creates and manages virtual disks. `vdiskadm` is implemented as a set of subcommands, some of which have their own options and operands. All operations on the `vdisk` need to be performed using `vdiskadm`.

The types of virtual disks are:

- `-t vmdk:fixed`
- `-t vmdk:sparse`
- `-t raw:fixed`
- `-t vdi:fixed`
- `-t vdi:sparse`
- `-t vhd:fixed`
- `-t vhd:sparse`

`vmdk` is the native VMware format, `vdi` is the native Sun VirtualBox format, `vhd` is the native Hyper-V format, and `raw` describes a file that looks like a raw disk. A raw disk is always in fixed format so that option can be explicitly set or implicitly understood. If the type is not specified, the default value is `vmdk`. If the option is not specified, the default value is fixed for type `raw` and sparse for types `vmdk` and `vdi`.

Create a new virtual disk of the specified size and at the location specified by `vdname`. If `vdname` includes a path to the virtual disk, the directories from that path will be created during creation of the virtual disk. `-t type[:opt],[opt]` specifies the type of virtual disk to be created. Type as one line.

```
# vdiskadm create -s size [-t type[:opt],[opt]] [-c comment] vdbname
```

You can import a disk image from a block device or file to a `vdisk`, convert it to a different type of `vdisk`, and export from a `vdisk` to a block device or file. This includes the full `vhd` support (sparse and fixed) and the ability to import a `vmdk` 1.1 optimized stream file. An optimized stream file is read-only and must be imported to another type (`vmdk:sparse` by default) in order to be used as a `vdisk`.
Examples

Creating a Default vmdk:sparse File

A directory of the vdisk name is created and populated with two files. vdisk.vmdk is the file with the disk data. vdisk.xml is the file containing information about the disk, such as creation time, type: option of disk, and snapshot information. Note that vdisk.vmdk has a suffix of the vdisk type.

```
# vdiskadm create -s 8g root_disk
# ls -l root_disk
total 82
-rw------- 1 root root 1114112 May 8 16:15 vdisk.vmdk
-rw-r--r-- 1 root root 584 May 8 16:15 vdisk.xml
```

Creating a vdisk File of Type vhd

The suffix specified is now vhd. Since the option isn’t specified with a type, the option has the default of sparse. Note that the disk file, vdisk_vhd, isn’t fully populated to 8G.

```
# vdiskadm create -s 8g -t vhd root_disk_vhd
# ls -l root_disk_vhd
total 44
-rw------- 1 root root 21504 May 8 16:15 vdisk.vhd
-rw-r--r-- 1 root root 590 May 8 16:15 vdisk.xml
```

Creating a vmdk:fixed File

Creating a vmdk type vdisk with option fixed takes a minute or more to create since it is creating and initializing 8G of data. The creation time is dependent upon the size of the vdisk.

```
# vdiskadm create -s 8g -t vmdk:fixed root_disk_fix
# ls -l root_disk_fix
total 16785428
-rw------- 1 root root 8589934592 May 8 16:18 vdisk-flat.vmdk
-rw------- 1 root root 638 May 8 16:18 vdisk.vmdk
-rw-r--r-- 1 root root 593 May 8 16:18 vdisk.xml
```

The contents of the xml file for root_disk_fix are:

```
# cat root_disk_fix/vdisk.xml
<?xml version="1.0"?>
<!DOCTYPE vdisk PUBLIC "-//Sun Microsystems Inc//DTD xVM Management All//EN" "file:///usr/share/lib/xml/dtd/vdisk.dtd">
<vdisk readonly="false" removable="false" cdrom="false" creation-time-epoch="1241821124" vtype="vmdk" sparse="false" rwcnt="0" rocnt="0">
  <name>root_disk_fix</name>
</vdisk>
```
This same information can be retrieved from the vdiskadm command by using the subcommand prop-get:

```
# vdiskadm prop-get -p all root_disk_fix
readonly: false
removable: false
cdrom: false
creation-time-epoch: 1241821124
vtype: vmdk
sparse: false
rwcnt: 0
rocnt: 0
name: root_disk_fix
version: 1.0
parent: none
filename: root_disk_fix
vdfile: vdisk.vmdk
owner: root
max-size: 8589934592
sectors: 16777216
description: none
effective-size: 8589935230
creation-time: Fri May 8 16:18:44 2009
modification-time: Fri May 8 16:18:44 2009
modification-time-epoch: 1241821124
```

The modification times and effective size are all derived “on the fly,” and are not stored in the xml file. The creation and modification times are shown both in epoch format and in human readable format, for use by both software applications (such as Sun Ops Center) and system administrators.

The rwcnt and rocnt fields shown in the xml file are the reader/writer locks on the vdisk. There can be only one writer at a time, but multiple readers can be using the vdisk. These fields are used to set or reset the reader/writer lock associated with the virtual disk. These fields should
not be set or reset by hand; they can only be modified by using `vdiskadm ref-inc [-r] vdame` or `vdiskadm ref-dec vdname`. These fields are used by `blktap` for shared or exclusive use of the virtual disk.

**Snapshots**

A snapshot is a read-only copy of a virtual disk. Snapshots can be created quickly and initially consume little space. As data within the active virtual disk changes, the snapshot consumes more data than would otherwise be shared with the active virtual disk.

`vdisk` supports snapshots in a manner similar to ZFS, except that the `vdisk` cannot be in use during a snapshot. The user can take a snapshot of the `vdisk` and later rollback to that snapshot, if needed. The user can also take a snapshot and then clone that snapshot into another `vdisk`.

To see the images are associated with a `vdisk`, type:

```
# vdiskadm list vhd_sp
vhd_sp
```

Take a snapshot of the virtual disk immediately after installing it:

```
# vdiskadm snapshot /export/home/vdisks/vhd_sp@install
```

List all images associated with the virtual disk:

```
# vdiskadm list /export/home/vdisks/vhd_sp
vhd_sp@install
vhd_sp
```

The original file, `vdisk.vhd`, has been moved to `vdisk@install.vhd`. A new file that contains the differences has been created. It is named `vdisk.vhd`.

```
# ls -l vhd_sp
total 2717732
-rw------- 1 root root 17408 May 11 16:41 vdisk.vhd
-rw-r--r-- 1 xvm root 717 May 11 16:41 vdisk.xml
-rw------- 1 root root 1390768640 May 11 16:41 vdisk@install.vhd
```

The `vdisk.xml` file shows the added snapshot element. When additional snapshots are created, new snapshot elements will be added to the XML description. The snapshot order in the list (and shown with `vdiskadm list`) shows the order in which the snapshots are loaded.

```
# cat vhd_sp/vdisk.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE vdisk PUBLIC "-/Sun Microsystems Inc//DTD xVM Management All//EN" "file:///usr/share/lib/xml/dtd/vdisk.dtd">
<vdisk readonly="false" removable="false" cdrom="false" creation-time-epoch="1241643718" vtype="vhd" sparse="true" rwcnt="0" rocnt="0">
```
Virtual Devices

Now, take another snapshot after a bfu and list the contents:

```
# vdiskadm snapshot /export/home/vdisks/vhd_sp@bfu

# vdiskadm list /export/home/vdisks/vhd_sp
vhd_sp@install
vhd_sp@bfu
vhd_sp
```

To roll back the disk to a point right after the install:

```
# vdiskadm rollback -r /export/home/vdisks/vhd_sp@install
# vdiskadm list /export/home/vdisks/vhd_sp
vhd_sp@install
vhd_sp
```

The rollback operation removes vdisk.vhd and any intervening snapshot images after vdisk@install.vhd, and creates a new differences file named vdisk.vhd.

```
# ls -l vhd_sp
total 2717732
-rw------- 1 root root 17408 May 11 16:47 vdisk.vhd
-rw-r--r-- 1 xvm root 717 May 11 16:47 vdisk.xml
-rw------- 1 root root 1390768640 May 11 16:47 vdisk@install.vhd
```

**Clones**

A clone is a writable copy of a virtual disk. The default type of clone is a merged (that is, coalesced) copy of the original virtual disk. An example of a merged clone occurs when a virtual disk is comprised of several snapshots; a subsequent clone operation results in a new virtual
disk containing no snapshots. A clone will be of the same type as the original virtual disk (that is, vmdk: fixed). When a merged clone is created there is no linkage back to the original virtual disk or to any of its snapshots. This lack of linkage allows the merged clone to be moved to another physical machine.

Create a clone of the specified snapshot or virtual disk. The clone is created with the type and option and the size of the virtual disk being cloned. If clone_vdname includes a path, the subdirectories from that path will be created during creation of the cloned virtual disk. By default, a merged clone image is created:

```
# vdiskadm clone [-c comment] vdname|snapshot clone_vdname
```

### About Domains

The control domain and the guest domain are separate entities.

Each domain has a name and a UUID. Domains can be renamed, but typically retain the same UUID.

A domain ID is an integer that is specific to a running instance. This ID changes whenever a guest domain is booted. A domain must be running to have a domain ID. UUID 0 is assigned to dom0.

### Control Domain 0

For the latest information on Domain 0, see the document [dom0 configuration for admins](http://www.opensolaris.org/os/community/xen/docs/configuring-dom0/).

The control domain is a version of Solaris modified to run under the xVM hypervisor. When the control domain is running, the control tools are enabled. In most other respects, the control domain 0 instance runs and behaves like an unmodified instance of the Solaris Operating System.

The control domain provides console access to the guest domains it controls, but you cannot otherwise access a guest domain from the control domain unless you use the remote login commands rlogin, telnet, and ssh. A control domain should be reserved for system management work associated with running a hypervisor. This means, for example, that users should not have logins on the control domain. The control domain provides shared access to a physical network interface to the guest domains, which have no direct access to physical devices.

If a control domain crashes with a standard Solaris panic, the dump will include just the control domain. Also see “About Crash Dumps” on page 546.
The following information applies to the control domain:

- ISA floppy is not supported.
- For 32-bit, the processor must support physical address extensions (PAE) mode.
- The OpenSolaris OS supports all Ethernet-type interfaces, and their data-links can be administered with the `dlaadm` command.

### Guest Domain Space Requirements

Size your domain as you would configure a machine to do the same workload.

The virtual disk requirement is dependent on the guest operating system and software that you install.

### Domain States

A domain can be in one of six states. States are shown in `virsh list` displays.

For example:

```bash
# virsh list

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Domain-0</td>
<td>running</td>
</tr>
<tr>
<td>2</td>
<td>sxc18</td>
<td>paused</td>
</tr>
</tbody>
</table>
```

The states are:

- **r**, running: The domain is currently running on a CPU.
- **b**, blocked: The domain is blocked, and not running or able to be run. This can be caused because the domain is waiting on IO (a traditional wait state) or it has gone to sleep because there was nothing running in it.
- **p**, paused: The domain has been paused, usually through the administrator running `virsh suspend`. When in a paused state, the domain still consumes allocated resources like memory, but is not eligible for scheduling by the hypervisor. Run `resume domain` to place the domain in the running state.
- **s**, in shutdown: The domain is in process of shutting down, but has not completely shutdown or crashed.
- **s**, shutoff: The domain is shut down.
The domain has crashed. Usually this state can only occur if the domain has been configured not to restart on crash. See `xmdomain.cfg(5)` for more information.
Booting and Running the Sun xVM Hypervisor

This chapter discusses booting and running the hypervisor.

x86 Boot Architecture

The open source GNU GRand Unified Bootloader (GRUB) is implemented on x86 based systems that are running the Solaris Operating System. GRUB is the boot loader that is responsible for loading a boot archive into a system’s memory. The boot archive contains the kernel modules and configuration files that are required to boot the system.

The GRUB main menu is based on the /boot/grub/menu.lst configuration file. For more information on GRUB, see Chapter 11, “Modifying Solaris Boot Behavior (Tasks),” in System Administration Guide: Basic Administration and grub(5).

Whether to run Solaris as a virtualized control domain or as a standalone operating system is a boot-time decision. To run the Solaris operating system as a standalone system, continue to use the same GRUB menu entries that you use currently.

In the Solaris xVM entry for booting Solaris as a control domain with the hypervisor, the kernel$ line refers to the hypervisor, and there are two module$ lines. The first module$ line must list the path to unix twice, with any arguments. The second module$ line lists the path to the boot archive.

In some situations, the GRUB menu.lst file might not reside in /boot/grub. To determine the location of the active GRUB menu.lst file, use the bootadm command with the list-menu subcommand.

For more information on the GNU GRand Unified Bootloader (GRUB) boot menu and its components, see System Administration Guide: Basic Administration.
Domain 0 Procedures

How to Set up OpenSolaris 2009.06 as a Sun xVM dom0

Use this procedure to set up the OpenSolaris 2008.11 or 2009.06 release as an xVM dom0.

1. Create a separate boot environment, if needed.

   ```
   # pfexec beadm create -a -d xvm xvm
   ```

2. If you’re running a recent version of OpenSolaris, you can install the xvm-gui cluster instead of the individual packages.
   Otherwise, install the required packages.

   ```
   # pfexec beadm mount xvm /tmp/xvm-be
   # pfexec pkg -R /tmp/xvm-be install SUNWvirt-manager SUNWxvm SUNWvdisk SUNWvncviewer
   # pfexec beadm umount xvm
   ```

3. Run the following AWK script over the menu.1st file that controls the GRUB menu. This step is necessary for beadm to reboot into xVM. The menu.1st file is typically located in the /pool-name/boot/grub directory on systems with a ZFS root. The correct menu.1st file is shown in Step 4 of this procedure.

   ```bash
   $ awk '/
   /^title/ { xvm=0; }
   /^title.xvm$/ { xvm=1; }
   /^(splashimage|foreground|background)/ { 
      if (xvm == 1) next
   }
   /^kernel\$/ { 
      if (xvm == 1) {
         print("kernel\$ /boot/$ISADIR/xen.gz")
         sub("^kernel\$", "modules")
         gsub("console=graphics", "console=text")
         gsub("i86pc", "i86xpv")
         $2=$2 " * $2"
      }
   }
   { print }' /rpool/boot/grub/menu.lst > /var/tmp/menu.lst.xvm
   ```

4. Verify that the AWK script worked properly.

   ```
   $ tail /var/tmp/menu.lst.xvm
   ```

End of LIBBE entry
Creating Guest Domain Environments by Using File Systems

Prior to installing guest domains (domain Us), you must decide how to create the guest domain environment.

- Use the Solaris ZFS file system and volume datasets to enable the rapid provisioning of guest domains.
- Use normal files to store guest domain disk images. Live migrations are done using Network File System (NFS).

### How to View Domains on the System

Run the `virsh list`, `xm list` or the `xm top` commands to view the domains on the system. These commands provide details of running domains. The display from any of these commands should show the single control domain, called Domain-0.

1. Become superuser, or assume the appropriate role.

2. Run `virsh list`.

   ```bash
   # virsh list
<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Domain-0</td>
<td>running</td>
</tr>
</tbody>
</table>
   ```

See Also: See the `xentop(1M)` man page.
How to Configure ZFS for Use With Solaris xVM

First, create a new storage pool using the `zpool` command described in `zpool(1M)`. The name of the disk can be specified as a full device path, such as `/dev/dsk/c0t0d0s0`, or as a disk name, such as `c1t0d0`. Multiple disks can be specified by using disk names separated by a space, to use disk striping.

Then, create a Solaris ZFS volume to store the guest domain master image. Note that the Solaris ZFS volume is a dataset, however, it represents a block device and can be used like traditional UNIX block devices.

1. Become superuser, or assume the appropriate role.

2. Create a new storage pool named `xpool` that includes the disk `c1t0d0`.
   
   ```bash
   # zpool create xpool c1t0d0
   ```
   
   The `-f` option can be used to force the action.

3. Verify that the storage pool is created:
   
   ```bash
   # zpool list
   ```
   
   ```
   NAME  SIZE USED  AVAIL CAP HEALTH ALTROOT
   xpool  34G  156K  34.0G  0%  ONLINE -L
   ```
   You can use the `zpool iostat` command to view information about the I/O throughput of the newly created storage pool.

4. Create an 8–Gbyte Solaris ZFS volume to store the guest domain master image.
   
   ```bash
   # zfs create -V 8gb xpool/domU-master
   ```

5. List the Solaris ZFS volume block device nodes, which are identified as devices in the `/dev/zvol/dsk` and `/dev/zvol/rdsk` directories.
   
   ```bash
   # ls -l /dev/zvol/dsk/xpool
   ```
   
   ```
   total 2
   lrwxrwxrwx 1 root 35 Apr 19 10:24 domu.master ->../../ll/ll/devices/pseudo/zfs@0:1c
   ```

How to Enable Migration to This Machine

This procedure is used to allow migration to this machine from all other control domains.

1. Become superuser, or assume the appropriate role.

2. Edit the `xend-relocation-address`, `xend-relocation-hosts-allow`, and `xend-relocation-server` properties of `xvm/xend`.
See Also  See dom0 configuration for admins
Creating and Managing Guest Domains

You can create, monitor, manage, and configure guests within a given OpenSolaris™ xVM hypervisor instance.

About Guest Installation

For the latest information on using the `virt-install` command-line utility to install guest domains, see *Using virt-install to Install a Guest*. This document provides xVM 3.3 examples.

To create a guest by using the `virt-install` utility, you must specify an installation source, disk storage, networking, and other parameters. After the guest OS is installed, it can be managed through the `virsh` utility, as described in "Managing Guests" on page 529.

Sizing Guests

The number of guests that can be created is determined primarily by the amount of memory and the disk space available.

Size your domain as you would configure a machine to do the same workload. The virtual disk requirement is dependent on the guest operating system and software that you install.

Installation Types

Types of `virt-install` installations that can be performed include the following:

- Interactive (not available in xVM 3.3)
- Command line, with options supplied
- Netinstall
- ISO image
About Network Installations

After you configure the installation server, you can run the `virt-install` command described in the `virt-install(1M)` from dom0. Use the `-d` option to `add_install_client` to specify that the client use DHCP. If the `-d` option is not specified, the client uses `bootparams`. For xVM paravirtualized guests, both approaches work. Use your site-specific tool for setting up the appropriate DHCP parameters for the client.

Network Installation

To do a network installation, use the `-l` option and provide a path to the network installation image. When giving a machine name or IP address, the domU must be able to get to that install machine directly, not go through a router to another net. For example:

```
-l nfs:install:/export/xvm/xvmgate-70i72-nd
```

You can also use an IP address instead of a machine name. For example:

```
-l nfs:172.20.25.12:/export/xvm/xvmgate-70i72-nd
```

**EXAMPLE 39-1** Network Installation

```
virt-install -n gath-01 -r 1000 --nographics -f /dev/dsk/c1t0d0s3 \
-m "aa:04:03:35:a8:06" -p \ 
-l nfs:install48:/export/xvm/xvmgate-70i72-nd
```

ISO Image Installation

To use the ISO image, use the `-l` option with a full path to the ISO image. If a full path is given instead of the `nfs:mach_name:path` format of a network installation, then `virt-install` assumes that this is an ISO image:

```
-l /net/install/export/xvm/solarisdvd.iso
```

**EXAMPLE 39-2** ISO Image

```
virt-install -n gath-01 -r 1000 --nographics -f /dev/dsk/c1t0d0s3 \ 
-m aa:04:03:35:a8:06 -p \ 
-l /net/install48/export/xvm/solarisdvd.iso
```
Note – You can quote arguments to options. While arguments, such as the path to an ISO image, are generally not quoted on the command line, quotes might be used in scripts.

- l "/net/install48/export/xvm/solarisdvd.iso"

JumpStart Installation
JumpStart configuration files are manually created and managed. You can initiate a custom JumpStart through network installation after setting up the server. When you create a profile server, you must ensure that systems can access the JumpStart directory on the profile server during a custom JumpStart installation. Each time that you add a system for network installation, use the add_install_client command to specify the profile server. You use the add_install_client command to create the /etc/bootparams entry for the domU.

To do a JumpStart with virt-install, use the --autocf option. For example:

--autocf nfs:install:/export/jumpstart/jump-user/x86

You cannot use a full path such as:

--autocf /net/install/export/jumpstart/jump-user/x86

EXAMPLE 39–3 JumpStart

virt-install -n gath-01 -r 1000 --nographics -f /dev/dsk/clt0d0s3 \ -m aa:03:35:a8:56 -p \ -l /net/install48/export/xvm/xvmgate-70i72/solarisdvd.iso
--autocf nfs:install:/export/jumpstart/jump-user/x86

Required Information
You will need to supply the guest domain information listed below.

name
Name for the guest domain. Each guest domain must have a unique name. This name serves as the label of the guest operating system. The name must be a real hostname for network installations to work.

image location
Location of the installation software. Installation must be over a network (which includes an NFS share from the local host operating system) or be an ISO install.

For example:

--location nfs:my.nfs.server.com:/home/install/test/mydomain
For HVM, an ISO or CDROM device should be given instead of an image location.

Installations using http or ftp, as shown in the following examples, are supported for Linux paravirtualized domain installations only:

http://my.http.server.com:/install/test/mydomain
ftp://my.ftp.server.com:/install/test/mydomain

vcpus
The number of CPUs for the guest domain. The default is 1. You can assign specific CPUs. If undefined, the hypervisor makes the selection.

memory
Amount of RAM to be allocated to the guest, in megabytes. A running domain should use a minimum of 512 megabytes. However, to install the guest domain, 1 Gbyte (1024 megabytes) is required.

graphics
Graphical console. Default is graphics. The nographics option applies to paravirtual guests only. If you intend to enable graphics support, you must decide whether the graphical installer should be used.

(Optional) Virtual network interface MAC address
This is the MAC address of the dom0’s network interface that you want the domU to use to send and receive internet traffic. By default, the hypervisor tools uses the first available network interface card (NIC) when creating guest domains.

The default values for the action to be taken on a domU shutdown, reboot, or crash are set by virt-install. You currently cannot change these defaults.

The complete list of supported virt-install options are listed in the virt-install(1M) man page.

Creating Guest Domains

Using virt-install to Install a Guest Domain

The virt-install program can be run as a command-line utility, with parameters specified through options, or interactively, in response to a series of prompts.

The default values for the action to be taken on a domU shutdown, reboot, or crash are set by virt-install. You currently cannot change these defaults.
Install Using `virt-install` With Options

This example uses `virt-install` with options to install a Solaris domU from the command line using an ISO image. The command line options specify for `virt-install` to create an 18-Gbyte root disk image file `/xvm/domu-x16.img`. The option `--nographics` is used because this is a Solaris paravirtualized configuration. If you invoke `virt-install` with command line options but do not supply all required information, the tool prompts you for the needed information.

```
machine:root> virt-install --nographics -n domu-x16 --paravirt \   -f /xvm/domu-x16.img -r 1011 \   -l /net/inst-server/export/xVM/x_iso/63-0419-nd.iso
Starting install...
Creating domain...
SunOS Release 5.11 Version 64-bit
Copyright 1983-2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Configuring /dev
Solaris Interactive Text (Console session)
Using install cd in /dev/dsk/c0d1p0
Using RPC Bootparams for network configuration information.
Attempting to configure interface xnf0...
Skipped interface xnf0
Setting up Java. Please wait...
Beginning system identification...
Searching for configuration file(s)...
Search complete.
Discovering additional network configuration...
```

When the domain creation completes, `sysidcfg` runs to complete the system identification.

Example: How to Install Open Solaris 2008.11 or Later in Paravirtualized Mode

Use this procedure to set up the OpenSolaris 2008.11 or later release as a paravirtual guest. You must be running a Solaris dom0 on your system.

1. To start the installation of the OpenSolaris 2008.11 or later release, run the following commands:

```
# zfs create rpool/zvol
# zfs create -V 10G rpool/zvol/domu-220-root
```
This procedure assumes that your server is set up to assign dynamic addresses. If you want to assign static addresses, specify the address with the `mac` property of the `-w/-n` network option. See limiting bandwidth and setting a VLAN ID.

2. Choose the defaults on the console for the two questions regarding the server setup.

3. After the OpenSolaris 2008.11 Live CD or OpenSolaris 2009.06 release has finished booting, a VNC session is available from within the guest domain. You can connect to the guest domain's VNC session as follows:

   ```
   # domid='virsh domid domu-220'
   # ip='/usr/lib/xen/bin/xenstore-read /local/domain/$domid/ipaddr/0'
   # port='/usr/lib/xen/bin/xenstore-read /local/domain/$domid/guest/vnc/port'
   # /usr/lib/xen/bin/xenstore-read /local/domain/$domid/guest/vnc/passwd
   DJP9tyDZ
   # vncviewer $ip:$port
   ```

4. Enter the given password at the VNC password prompt to bring up a VNC session.

   VNC sessions are not secure. However, because you need elevated privileges to read the VNC password from XenStore, the VNC sessions are secure as long as you run the VNC viewer locally on dom0, or via SSH tunnelling or another secure method.

5. Enable the post-install VNC viewer.

   By default, the VNC session is not enabled after the installation. You can change the default configuration as follows:

   ```
   # svccfg -s x11-server setprop options/xvm_vnc = "true"
   # svcadm restart xvm/vnc-config
   # svcadm restart gdm
   ```

### Example: How to Install and Boot a Red Hat Enterprise Linux 5.3 PV Guest on an OpenSolaris 2009.06 dom0

The following sequence of commands installs a Red Hat Enterprise Linux guest over NFS using the text installer:

```
# mount -F hsfs /rhel.iso /mnt
# share -o ro /mnt
# virt-install -n pv-rhel -r 1024 -l nfs:mydom0:/mnt \
   --os-type=linux os-variant=rhel5.3 \ 
   -f /dev/zvol/dsk/pv-rhel.zvol -p --nographics
```

The following command installs a Red Hat Enterprise Linux guest using the media in the dom0 optical drive (CD-ROM/DVD), utilizing the RedHat Linux version 5 kickstart feature to automate the installation process.
Because of a standard Linux restriction that PV guests cannot be installed from a CD, the CD must be mounted in a location (usually dom0) exported over NFS, and then an NFS Linux installation done. Often it is much easier to do an HTTP Linux install.

**Additional `virt-install` Examples**

**EXAMPLE 39–4** Solaris PV Guest

```bash
# virt-install \
   --name rhat \ 
   --ram 500 \ 
   --file /dev/zvol/dsk/rhat.zvol \ 
   --paravirt \ 
   --location /dev/dsk/c2t0d0s2 \ 
   --os-type=linux os-variant=rhel5 \ 
   --extra-args "ks=/export/install/rhat/ks.cfg"
```

**EXAMPLE 39–5** Solaris HVM Guest

```bash
virt-install -n solarisHVM --hvm -r 1024 --vnc \ 
   -f /export/solarisHVM/root.img -s 16 \ 
   -c /ws/xvm-gate/public/isos/72-0910/solarisdvd.iso
```

For this version of `virt-install`, ISO, physical media, and Preboot Execution Environment (PXE) pxe network installations are supported for HVM. If a physical CD is used, remember to unmount it after use.

**EXAMPLE 39–6** Windows HVM Guest

```bash
# virt-install -n winxp --hvm -r 1024 --vnc \ 
   -f /export/winxp/root.img -s 16 -c /windows/media.iso
```

For this version of `virt-install`, ISO, physical CD, and PXE network installations are supported for HVM. If a physical CD is used, remember to unmount it after use.

**EXAMPLE 39–7** Install Microsoft Windows Using a Local File as a Root Disk

A normal file is used to store the contents of the guest domain disk image, as opposed to using a ZFS volume, for example.
EXAMPLE 39–7  Install Microsoft Windows Using a Local File as a Root Disk  (Continued)

    virt-install --name windows1 --ram 1024  \
    --cdrom /en_winxp_pro_with_sp2.iso --file /guests/windows1-disk  \
    --file-size 10 --vnc

EXAMPLE 39–8  Install Solaris Using Network Install and JumpStart onto a ZFS Volume

    zfs create -V 8G pool/solaris1-disk
    virt-install --name solaris1 --ram 1024 --nographics  \
    --file /dev/zvol/dsk/pool/solaris1-disk  \
    --location nfs:install.domain.com:/export/solaris/nv75  \
    --autocf nfs:install.domain.com:/export/jumpstart/solaris1

▼ How to Complete the Solaris DomU sysidcfg Configuration

● After the domain is created, the sysidcfg is initiated and you are prompted to answer a series of questions. Your screen will look similar to this:

SunOS Release 5.11 Version 64-bit
Copyright 1983-2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Hostname: my-zone
Loading smf(5) service descriptions: 114/114
Select a Language

  1. English
  2. French
  3. German
  4. Italian
  5. Japanese
  6. Korean
  7. Simplified Chinese
  8. Spanish
  9. Swedish
 10. Traditional Chinese

Please make a choice (1 - 10), or press h or ? for help:

What type of terminal are you using?
  1) ANSI Standard CRT
  2) DEC VT52
  3) DEC VT100
4) Heathkit 19
5) Lear Siegler ADM31
6) PC Console
7) Sun Command Tool
8) Sun Workstation
9) Televideo 910
10) Televideo 925
11) Wyse Model 50
12) X Terminal Emulator (xterms)
13) CDE Terminal Emulator (dtterm)
14) Other

Type the number of your choice and press Return:

For more information on the `sysidcfg` file, see the `sysidcfg(4)` man page. For an example `sysidcfg` file, see *Understanding the Solaris xVM Server Architecture*, Part No 820-3089-102.

**Managing Guests**

The main interface for command and control of both xVM control domains and guest domains is the `virsh(1M)` utility. Users should use `virsh` wherever possible to control virtualized operating systems. Some xVM operations are not yet implemented by the `virsh` utility. In those cases, the legacy utility `xm(1M)` can be used for detailed control.

The following actions can be performed:

- Start Selected Guest
  ```
  # virsh start sdomu
  ```
- Suspend Selected Guest
  ```
  # virsh suspend sdomu
  ```
  You can also use the `destroy` subcommand to bring the system to the shutoff state, but this can result in damage to data.
- Resume Selected Guest
  ```
  # virsh resume sdomu
  ```
- Shutdown Selected Guest
  ```
  # virsh shutdown sdomu
  ```
- Reboot Selected Guest
  ```
  # virsh reboot sdomu
  ```
Creating Guest Domains

- **Undefine Selected Guest**
  
  ```bash
  # virsh undefine sdomu
  ```

  Undefine the configuration for an inactive domain which is specified by either its domain name or UUID.

  To delete a guest, undefine the guest and then remove any associated disk resources.

- **Connect Selected Guest to Network**
  
  ```bash
  # virsh attach-interface sdomu bridge e1000g0
  ```

- **Take Snapshot of Active Domain**
  
  ```bash
  # virsh save sdomu /domains/sdomusnap
  ```

  The domain will be in the shut down state when the save operation completes. The resources, such as memory, allocated for the domain will be freed for use by other running domains.

  To restore:

  ```bash
  # virsh restore sdomu /domains/sdomusnap
  ```

  Note that network connections present before the save operation might be severed, since TCP timeouts might have expired.

- **Pin vCPU**
  
  ```bash
  # virsh vcpupin domain vcpu cpulist
  ```

  Pin domain's virtual CPUs to the host's physical CPUs. The `domain` parameter is the domain name, ID, or UUID. The `vcpu` parameter is the VCPU number. The `cpulist` parameter is a list of host CPU numbers, separated by commas.

- **Restore Selected Guest**
  
  ```bash
  # virsh restore sdomu
  ```

  Restore a domain that is in the saved state.

- **Delete Selected Guest**
  
  Deleting the guest will cause its image and snapshot to be deleted from the system.

To view the state of a domain, use the `virsh list` command.

For example:
By default, only running domains are displayed. Use `virsh list --inactive` to display only non-running domains.

### Live Migration

The following prerequisites apply:

- Both the source machine and the target host must be on the same subnet.
- The host and the target must each have the same CPU type (AMD or Intel).
- Both systems must be running the same release of the xVM software.
- There must be sufficient CPU and memory resources on the target to host the domain.
- The target dom0 should have the same network interface as the source dom0 network interface used by the domU. For example, if the domU to be migrated has a VNIC that is bridged over the e1000g0 interface on the source dom0, then the target dom0 must also have the e1000g0 interface.

#### Enabling Live Migration on a Target Host

By default, xend listens only on the loopback address for requests from the `localhost`. The target host must be configured to accept the migration of a guest domain. The following example configures the xend SMF service on the target machine to accept guest migration from a system named `host1`. The caret (\(^\)) and dollar sign (\($\)) are pattern-matching characters to ensure that the entire host name matches. The `host1` name must match the name the target thinks the machine is called, which could be a host name, but could also be a fully qualified domain name (FQDN).

```
# svcadm refresh svc:/system/xvm/xend:default && 
svcadm restart svc:/system/xvm/xend:default
```

You can test the connection by using:

```
host1# telnet target-host 8002
```

If connection fails, check the `/var/log/xen/xend.log` file on the target system.
Configuring the Guest Domain to Be Migrated

In addition to configuring the target system to accept migrations, you must also configure the domain that will be migrated so that the domain’s storage is accessible from both the source and the target systems. The domain’s accessible disks must reside on some form of shared storage, such as NFS files or iSCSI volumes. This document uses the NFS method available in the OpenSolaris 2009.06 release.

Prepare the NFS Storage on the NFS Server

On the NFS server, share the directory:

```
# sharectl set -p nfsmapid_domain=sun.com nfs
# svcadm restart svc:/network/nfs/mapid:default
# share -f nfs -o "sec=sys,root=host1:host2,rw" /vdisks
```

On both the host1 source system and the host2 target system, also execute the sharectl to set the NFS mapid name to sun.com, and the svcadm command restart the xend service.

Create a PV Guest on the Source Machine (host1)

```
# virt-install -p --nographics -n domain -r 1024 -l /isos/os0906/os0906.iso -f /net/hostname_of_nfs_server/vdisks/testpv
```

The virt-install command then creates the virtual disk on the NFS server and starts the guest installation process.

iscsi Volume Method, Available in a Later Build

Note – This method should be available in a build after the OpenSolaris 2009.06 release.

An iscsi-backed guest would be created with this base command:

```
# virt-install -n <name> -r <ram> -p --nographics -l /path/to/iso \
-m <mac address> \ 
--disk path=/static/<iscsi target ip address>/<lun>/<iqnnumber>,driver=phy,subdriver=iscsi
```

An example would be:

```
# virt-install -n ubuntu -r 1024 -p --nographics -l /net/xvm-4200m2-03/isos/ubuntu-7.04-32.iso \ 
--disk path=/static/172.20.26.10/0/iqn.1986-03.com.sun:02:52ac879e-788e-e0ea-bf5c-f86b2b63258a,driver=phy,subdriver=iscsi
```

In addition to setting up the xend relocation host allow field as described above in the section "Enabling Live Migration on a Target Host," also issue the following command to enable static discovery on both dom0s:

```
# iscsidadm modify discovery -s enable
```
Migrating the Guest Domain

Use the following command to migrate the guest.

```
host1# virsh migrate domain --live xen:// xenmigr://target-host
```

Note – While the use of the `virsh` command is preferred, you can try the following command if the `virsh` command appears to fail.

```
host1# xm migrate -l domain target-host
```

You can observe the migration while it occurs by monitoring the domain status on both machines using `virsh list`.

Note that the domain definition remains on the system on which it was created. You can start the domain on that system's dom0 with the `virsh start domain` command.

Assigning a CD-ROM Device to a Guest

Assigning the CD-ROM Device to an HVM Guest

You cannot add temporary access to a device for an HVM domU because you cannot dynamically add IDE CD-ROM devices. To make a CD-ROM available for use in an HVM guest, you must add the device before you boot the guest. You can then change the media by using `virsh attach-disk`.

Note that in an HVM domU, you must use `eject` in the domU to unlock a removable-media device, such as a CD device, before running the `attach-disk` subcommand.

The `attach-disk` subcommand is also used to change the media in a CD drive.

For additional explanation, see the `virsh(1M)` man page, including “Example 4 Changing a CD in a Solaris HVM Guest Domain.”

Assigning the CD-ROM Device to a PV Guest

In PV guest domains, you can use `virsh attach-disk` to add a CD device and `detach-disk` to remove it. These commands can be used on halted or running domUs. If you want to change the CD after attaching the device, you must use `virsh detach-disk` and then `virsh attach-disk` with the new media.
About Xvnc

Virtual network computing (VNC) is a remote control software product that allows you to view and fully interact with one computer desktop, the Xvnc server, by using the VNC viewer on another computer desktop. The two computers do not have to be running the same type of operating system. VNC provides a guest domain graphical login.

By default, consoles for HVM guests are graphics consoles. You can use VNC to view a Windows guest domain from a Solaris dom0. You only need to set the address and password for VNC to work with HVM guests. HVM installs may specify either VNC (-vnc) or Simple DirectMedia Layer (SDL) (-sdl) for graphics support. You can later configure the OS to use the serial console as the main console.

Use the vncpasswd command to set the password used to access VNC desktops. The password is stored on the server. For more information, see vncpasswd(1).

Xvnc displays to a VNC viewer over the network. The VNC server display number is the same as the X server display number. For example, snoopy:2 refers to display 2 on machine snoopy for both VNC and an X server.

For the latest information on VNC setup, see (http://www.opensolaris.org/os/community/xen/docs/setupvnc/).

How to Set Up VNC to Provide a Solaris Guest Graphical Login with CDE

1 Become superuser, or assume the appropriate role.

2 Enable XDMCP connections:
   
   # svccfg -s cde-login
cvc:/application/graphical-login/cde-login> setprop dtlogin/args=""

3 (Optional) If you are not running vncviewer locally on the control domain, set X11-server to listen to the tcp port:

   # svccfg -s x11-server
   svc:/application/x-11/x11-server> setprop options/tcp_listen=true

   The VNC listen facility should be used with caution due to security considerations.

4 Enable the Xvnc inetd services.

   # svcadm enable xvnc-inetd
How to Set Up VNC to Provide a Solaris Guest Graphical Login with GDM

1. Become superuser, or assume the appropriate role.

2. Enable XDMCP for GDM:
   ```
   # printf 'xdmcp
   # Enable=true
   ' | tee /etc/X11/gdm/custom.conf
   # svcadm restart gdm
   ```

3. Make sure that GDM is running:
   ```
   # svcadm enable -s gdm
   ```

4. Set the X11-server to listen to the tcp port:
   ```
   # svcadm -s x11-server
   svc:/application/x-11/x11-server> setprop options/tcp_listen=true
   ```

5. Enable the Xvnc inetd services:
   ```
   # svcadm enable xvnc/inetd
   ```

6. Connect from another machine and verify that you see the login screen and can log in to a desktop session.
   ```
   # vncviewer domU:0
   ```

How to Start VNC at System Boot

This procedure starts VNC at system boot from the dtlogin, displaying the dtlogin login screen.

1. Become superuser, or assume the appropriate role.

2. Add an instance of x11-server service called display1 for configuration, and configure it to run Xvnc.
   ```
   svccfg -s application/x11/x11-server add display1
   svccfg -s application/x11/x11-server:display1 addpg options application
   svccfg -s application/x11/x11-server:display1 addpropvalue options/server astring: "/usr/X11/bin/Xvnc"
   ```
Configure dtlogin to start it.

```bash
svccfg -s application/x11/x11-server:display1 addpropvalue options/server_args astring: "SecurityTypes=None"
```

3. Configure dtlogin to start it.

```bash
mkdir -p /etc/dt/config
cp /usr/dt/config/Xservers /etc/dt/config/Xservers
echo " :1 Local local_uid@none root /usr/X11/bin/Xserver :1" >> /etc/dt/config/Xservers
pkill -HUP dtlogin
```

4. Connect from another machine and verify that you see the login screen and can log in to a desktop session.

```bash
# vncviewer domU:0
```

### How to Start a GNOME Session

- Use the following to start the GNOME session.

```bash
# /bin/sh
# mkdir <your homedir>/.vnc
# echo "#!/bin/sh
/usr/bin/dbus-launch /usr/bin/gnome-session" > <your homedir>/.vnc/xstartup
```

### How to View the VNC Man Pages

You can use the man command to view the man pages.

- man Xvnc
- man vncviewer
- man vncpasswd

**Note** – Live links to these man pages cannot be made from this book.

The MANPATH variable is normally set for you in desktop login sessions. If the entry is not found, check your MANPATH environment variable and add the path to the X11 manpages if necessary.

1. **View the MANPATH:**

   ```bash
   echo $MANPATH
   /usr/share/man:/usr/dt/man:/usr/openwin/share/man
   ```

2. **If necessary, add the path to the X11 man pages.**

   ```bash
   setenv MANPATH /usr/share/man:/usr/dt/man:/usr/openwin/share/man:/usr/X11/man
   ```
This chapter covers xVM system administration topics.

**Printing Kernel and Machine Information**

Use `uname` to determine the kernel you are running.

```
hostname% uname -a
SunOS hostname 5.11 snv_80 i86pc i86 i86xpv
```

Use the `isainfo` command to print the basic application environments supported by the currently running system.

```
hostname% isainfo -x
amd64: sse2 sse fxsr amd_3dnowx amd_3dnow amd_mmx mmx cmov cx8 tsc fpu
i386: ahf sse2 sse fxsr amd_3dnowx amd_3dnow amd_mmx mmx cmov cx8 tsc fpu
```

Use the `psrinfo` command to display information about processors.

```
hostname% psrinfo -vp
The physical processor has 1 virtual processor (0)
x86 (Authentic AMD family 15 model 5 step 10 clock 2200 MHz)
Dual Core AMD Opteron(tm) Processor 275
```
Configuring the Serial Console to Be the Main Console

After an HVM guest domain has been installed, you can configure the OS to use the serial console as the main console.

▼ How to Configure the Serial Console as the Main Console

1. Type:
   ```
   # eeprom console=ttya
   ```

2. Reboot the HVM guest domain.

virsh Command and Domain Management

The main command interface used to control both Solaris xVM and guest domains is the `virsh` command. `virsh` provides a generic and stable interface for controlling virtualized operating systems. Use `virsh` instead of `xm` wherever possible.

Many `virsh` commands act asynchronously. This means that the system prompt can return before the operation has completed.

If you modify CPUs or memory by using the `virsh` command, these changes will be saved in the configuration file and persist across reboots.

virsh Command Structure

Most `virsh` commands follow the format:

```
# virsh subcommand domain-id | name | uuid [options]
```

- `subcommand` One of the subcommands listed in the `virsh(1M)` man page
- `domain-id, name, or uuid` A domain identifier
- `options` An option to a subcommand

EXAMPLE 40-1 Using a virsh Command

This line connects to a domU named `sxc18`.

```
# virsh console sxc18
```
The `virsh` is used to manage domains. You must run the commands as the root user or by assuming the appropriate role account on the host operating system. The commands cannot be run in the guest domain.

<table>
<thead>
<tr>
<th><code>virsh</code> Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>virsh attach-device</code></td>
<td>Attach device from an XML file</td>
</tr>
<tr>
<td><code>virsh attach-disk</code></td>
<td>Attach disk device</td>
</tr>
<tr>
<td><code>virsh autostart</code></td>
<td>Configure a domain to automatically start at boot time.</td>
</tr>
<tr>
<td><code>virsh capabilities</code></td>
<td>Return capabilities of the hypervisor and drivers.</td>
</tr>
<tr>
<td><code>virsh connect</code></td>
<td>Connect to the hypervisor</td>
</tr>
<tr>
<td><code>virsh connect--readonly</code></td>
<td>Connect to the hypervisor in read-only mode.</td>
</tr>
<tr>
<td><code>virsh console domain</code></td>
<td>Connect to a guest console</td>
</tr>
<tr>
<td><code>virsh create file</code></td>
<td>Create a domain based on the parameters contained in the XML file, where file is an absolute pathname. Such a file can be created using <code>virsh dumpxml</code> subcommand. The XML configuration file should not be directly edited.</td>
</tr>
<tr>
<td><code>virsh define file</code></td>
<td>Define a domain from an XML file, but do not start the domain</td>
</tr>
<tr>
<td><code>virsh destroy domain-id</code></td>
<td>Terminate a domain immediately</td>
</tr>
<tr>
<td><code>virsh detach -device domain-id file</code></td>
<td>Detach a device defined by the given XML file from the specified domain.</td>
</tr>
<tr>
<td><code>virsh domid domain_name</code></td>
<td>Converts a domain name to a numeric domain ID.</td>
</tr>
<tr>
<td><code>virsh dominfo domain_id</code></td>
<td>Return basic info about a domain</td>
</tr>
<tr>
<td><code>virsh domname domain_id</code></td>
<td>Converts a numeric domain ID to a domain name.</td>
</tr>
<tr>
<td><code>virsh domstate domain_id</code></td>
<td>Returns the state of a running domain. See the list subcommand.</td>
</tr>
<tr>
<td><code>virsh domuuid domain</code></td>
<td>Convert the specified domain name or ID to a domain UUID.</td>
</tr>
<tr>
<td><code>virsh dump domainfile</code></td>
<td>Dump the core of the domain specified by domain to the file specified by file for analysis.</td>
</tr>
<tr>
<td><code>virsh dumpxml domain-id</code></td>
<td>Obtain domain information in XML</td>
</tr>
<tr>
<td><strong>virsh Command</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><strong>virsh help</strong></td>
<td>Display descriptions of the subcommands. Include a subcommand at the end of the command line to display help about that subcommand.</td>
</tr>
<tr>
<td><strong>virsh list</strong></td>
<td>List domains. By default, only running domains are displayed. Use --inactive to display only non-running domains.</td>
</tr>
<tr>
<td><strong>virsh nodeinfo</strong></td>
<td>Print basic information about a node.</td>
</tr>
<tr>
<td></td>
<td># virsh nodeinfo</td>
</tr>
<tr>
<td></td>
<td>CPU model: 186pc</td>
</tr>
<tr>
<td></td>
<td>CPU(s): 2</td>
</tr>
<tr>
<td></td>
<td>CPU frequency: 2391 MHz</td>
</tr>
<tr>
<td></td>
<td>CPU socket(s): 2</td>
</tr>
<tr>
<td></td>
<td>Core(s) per socket: 1</td>
</tr>
<tr>
<td></td>
<td>Thread(s) per core: 1</td>
</tr>
<tr>
<td></td>
<td>NUMA cell(s): 1</td>
</tr>
<tr>
<td></td>
<td>Memory size: 4127744 kB</td>
</tr>
<tr>
<td><strong>virsh quit</strong></td>
<td>Quit this interactive terminal</td>
</tr>
<tr>
<td><strong>virsh reboot domain-id</strong></td>
<td>Reboot a domain.</td>
</tr>
<tr>
<td></td>
<td>This command is identical to the effect of running init 6. The command returns immediately, but the entire reboot process might take a minute or more.</td>
</tr>
<tr>
<td><strong>virsh restore state-file</strong></td>
<td>Restore a domain from a saved state file.</td>
</tr>
<tr>
<td><strong>virsh resume domain-id</strong></td>
<td>Moves a domain out of the paused state, making the domain eligible for scheduling by the hypervisor.</td>
</tr>
<tr>
<td><strong>virsh reboot domain-id</strong></td>
<td>Reboot a domain</td>
</tr>
<tr>
<td><strong>virsh restore domain-id</strong></td>
<td>Restore a domain from a saved state</td>
</tr>
<tr>
<td><strong>virsh resume domain-id</strong></td>
<td>Resume running a suspended domain.</td>
</tr>
<tr>
<td><strong>virsh save domain state-file</strong></td>
<td>Save a running domain to a state file so that it can be restored by using the restore subcommand at a later time. In this state, the domain is not running on the system, thus the memory allocated for the domain will be free for use by other domains.</td>
</tr>
<tr>
<td></td>
<td>Note that network connections present before the save operation might be severed because TCP timeouts might have expired.</td>
</tr>
<tr>
<td><strong>virsh setvcpus domaincount</strong></td>
<td>Change the number of virtual CPUs active in the specified guest domain. The count parameter is the number of virtual CPUs.</td>
</tr>
<tr>
<td><strong>virsh Command</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>virsh schedinfo domain</strong></td>
<td>Show or set the scheduling parameters for the specified domain name, ID, or UUID. This subcommand takes the options <code>-w</code> <code>weight</code> number and <code>-c</code> <code>cap</code> number.</td>
</tr>
<tr>
<td><strong>virsh setmaxmem domain kilobytes</strong></td>
<td>Change the maximum memory allocation limit in the specified guest domain. The <code>kilobytes</code> parameter is the maximum memory limit in kilobytes.</td>
</tr>
<tr>
<td><strong>virsh setmem domain kilobytes</strong></td>
<td>Change the current memory allocation in the specified guest domain. The <code>kilobytes</code> parameter is the number of kilobytes in memory.</td>
</tr>
<tr>
<td><strong>virsh setvcpus domain count</strong></td>
<td>Change the number of virtual CPUs active in the specified guest domain. The <code>count</code> parameter is the number of virtual CPUs.</td>
</tr>
<tr>
<td><strong>virsh shutdown domain</strong></td>
<td>Coordinates with the domain operating system to perform graceful shutdown. The effect of this command is identical to the effect of running <code>init 5</code>. The shutdown might take an unexpected length of time, depending on the services that must be shut down in the domain. In addition, it is possible that the subcommand will not succeed.</td>
</tr>
<tr>
<td><strong>virsh start domain</strong></td>
<td>Start a previously defined inactive domain.</td>
</tr>
<tr>
<td><strong>virsh suspend domain</strong></td>
<td>Suspend a domain. A domain in this state still consumes allocated resources, such as memory, but is not eligible for scheduling by the hypervisor.</td>
</tr>
<tr>
<td><strong>virsh undefined domain</strong></td>
<td>Undefine the configuration for the inactive domain by specifying either its domain name or UUID.</td>
</tr>
<tr>
<td><strong>virsh vcpuinfo domain</strong></td>
<td>Return basic information about the domain's virtual CPUs.</td>
</tr>
<tr>
<td><strong>virsh vcpupin domain vcpu cpulist</strong></td>
<td>Pin domain's virtual CPUs to the host's physical CPUs. The <code>domain</code> parameter is the domain name, ID, or UUID. The <code>vcpu</code> parameter is the VCPU number. The <code>cpulist</code> parameter is a list of host CPU numbers, separated by commas.</td>
</tr>
<tr>
<td><strong>virsh version</strong></td>
<td>Display version information.</td>
</tr>
<tr>
<td><strong>virsh vncdisplay domain-id</strong></td>
<td>VNC display</td>
</tr>
</tbody>
</table>
**Ethernet-Type Interface Support**

The OpenSolaris™ OS supports all Ethernet-type interfaces, and their data-links can be administered with the `dladm` command.

**Suspend and Resume Functions and Commands**

Some xVM operations are not yet implemented in the `virsh` command. In those cases, the equivalent `xm` command can be used. Subcommand terminology differs between the `xm` and `virsh` commands. In particular, the `suspend` and `resume` commands have different meanings.

<table>
<thead>
<tr>
<th><code>virsh</code></th>
<th><code>xm</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>suspend</td>
<td>pause</td>
</tr>
<tr>
<td>resume</td>
<td>unpause</td>
</tr>
<tr>
<td>save</td>
<td>suspend without an output file argument</td>
</tr>
<tr>
<td>restore</td>
<td>resume without an output file argument</td>
</tr>
</tbody>
</table>

**Cloning ZFS-Based Solaris Domains**

If you are using a ZFS volume as the root disk for a domU, you can use the ZFS snapshot facilities to clone another domU with the same configuration. By taking a clone of the root disk, you can quickly provision similar domains.

For example, you might install Solaris as a guest domain, run `sys-unconfig(1M)`, then clone that disk image for use in new Solaris domains. Installing a Solaris domain in this way requires only the configuration step, rather than a full install. The only extra storage used for the cloned domain is the amount needed for the differences between the domains.

You also have the capability to revert to a previous configuration if necessary.

---

**Note** – Any clones created from a snapshot must be destroyed before the snapshot can be destroyed.

---

**How to Use ZFS Snapshot to Clone a Solaris DomU**

If you use a ZFS volume as the virtual disk for your guest domain, you can take a snapshot of the storage. The snapshot is used to create clones.
Note that you might want to use the `sys-unconfig` command described in *sys-unconfig*(1M) in the domain before you take the snapshot. The resulting clones would not have host names or name services configured, which is also known as "as-manufactured." When it comes up, the new clone displays the configuration screens.

1. Become superuser, or assume the appropriate role.

2. (Optional) To create a snapshot to produce domains that require `sysidcfg` to complete system identification, use the `sys-unconfig` command in a domain named `domain1`.

   
   ```
   # virsh shutdown domain1
   ```

4. Take a snapshot of the root disk used by `domain1`.
   
   ```
   # zfs snapshot pool/domain1-root@clone
   ```

5. Create a clone named `domain2` from the snapshot `domain1-root@clone`.
   
   ```
   # zfs clone pool/domain1-root@clone pool/domain2-root
   ```

6. (Optional) Display the snapshot and clone.
   
   ```
   # zfs list
   # virsh dumpxml domain1 >domain1.xml
   # cp domain1.xml domain2.xml
   ```

7. Dump the configuration of `domain1`.
   
   ```
   # virsh dumpxml domain1 >domain1.xml
   ```

8. Copy the configuration file `domain1.xml` to a file named `domain2.xml`.
   
   ```
   # cp domain1.xml domain2.xml
   ```

9. Make the following changes in the `domain2.xml` file.
   
   a. Replace `domain1` in this line:
      
      ```
      <name>domain1</name>
      ```
      
      With the new name, `domain2`:
      
      ```
      <name>domain2</name>
      ```

   b. So that `virsh` will generate a new domain configuration, remove the UUID line, which looks like this:
      
      ```
      <uuid>72bb96b6e6cf13594fb0cd1290610611</uuid>
      ```
c. Point to the new disk by editing the following line:

```xml
<source dev='/dev/zvol/dsk/export/domain1-root'/>
```

Change domain1-root to domain2-root so that the line appears as follows:

```xml
<source dev='/dev/zvol/dsk/export/domain2-root'/>
```

10 Inform `virsh` about the new domain:

```bash
# virsh define domain2.xml
```

11 Boot the cloned domain.

**More Information**

**More Information on ZFS Snapshot Features**

Also see Chapter 7, “Working With ZFS Snapshots and Clones,” in *Solaris ZFS Administration Guide*.

**Recovery**

You can keep snapshots of the guest domain OS installations that are known to be good images, and use ZFS rollback to revert to a snapshot if the domain has a problem. For more information, see “Rolling Back a ZFS Snapshot” in *Solaris ZFS Administration Guide*.

**Communication From xVM Hypervisor to Dom0 Using `xm`**

Although the hypervisor and dom0 work closely together to manage a running system, the dom0 operating system has little direct visibility into the hypervisor. The hypervisor’s entire address space is inaccessible to the dom0.

The only source of information is provided by the `xm` command, a user-space tool that communicates with the hypervisor via hypercalls.

Some of the commonly used `xm` commands are:

- `xm info`  
  Report static information about the machine, such as number of CPUs, total memory, and xVM version.

```
# xm info
host : test
release : 5.11
version : onnv-userj
machine : i86pc
nr_cpus : 2
nr_nodes : 1
```
sockets_per_node         : 2
cores_per_socket         : 1
threads_per_core         : 1
cpu_mhz                 : 2391
hw_caps                 : 078bfbff:e1d3fbff:00000000:0000000010
total_memory            : 4031
free_memory             : 1953
xen_major               : 3
xen_minor               : 1
xen_extra               : .2-xvm
xen_caps                : xen-3.0-x86_64 xen-3.0-x86_32p
xen_scheduler           : credit
xenpagesize             : 4096
platform_params         : virt_start=0xffff800000000000
xen_changeset           : Thu Dec 20 20:11:49 2007 -0800 15623:41d827ccee7
cc_compiler             : gcc version 3.4.3 (csl-sol210-3_4-20050802)
cr_compile_by           : userj
cr_compile_domain       : lab.sun.com
cr_compile_date         : Thu Dec 20 20:24:36 PST 2007
xend_config_format      : 4

xm list            List all domains and some high-level information.
xm top             Analogous to the Linux top command, but it reports domain
                    information instead of process information. Information about the xVM
                    system and domains is displayed in a continuously updating manner
                    through the xentop command. See xentop.
xm log             Display the contents of the xend log.
xm help            List all the available commands.
xentrace           Capture trace buffer data from xVM.
xentop             Display information about the xVM system and domains in a
                    continuously updating manner. See xm top.
xm start domain    Start a managed domain that was created by virt-install.

If you modify guest domain CPUs or memory by using the xm command, these changes will be
saved in the configuration file and persist across reboots.

See the xm(1M) man page for more information.
Domain 0 and Hypervisor Crashes

On a running system, the hypervisor’s memory is completely off-limits to dom0. If the hypervisor crashes, however, the resulting panic dump will generate a core file that provides a unified view of both xVM and dom0. In this core file, xVM appears as a Solaris kernel module named xpv. For example:

```
$ sc
   xpv'panic+0xbf()
   xpv'do_crashdump_trigger+0x19()
   xpv'keypress_softirq+0x35()
   xpv'do_softirq+0x54()
   xpv'idle_loop+0x55()
```

The following applies to crash dumps:

- If a dom0 crashes with a standard Solaris panic, the dump will include only the dom0.
- When the hypervisor itself panics, the resulting dump includes the xVM state as well as the dom0 state.

How to Force a Crash Dump of a Guest

You can use the following command to force a crash dump of an OpenSolaris guest in the event of problems within the guest:

```
# virsh dump domain
```

The crash dump file created can be analyzed through /bin/mdb. The xvm user must be able to write to the location specified.
Troubleshooting Miscellaneous Sun xVM Problems

This chapter contains troubleshooting information for Sun xVM.

Dom0 Configuration Requirements

At this time, you should note the following issues when configuring dom0:

ZFS ARC limitation needed

ZFS must be prevented from taking too much memory from dom0. Limit the ARC by adding the following line to /etc/system:

```
set zfs:zfs_arc_max = 0x10000000
```

Reboot the system.

Limit dom0 memory

Limit dom0 to a specific amount of memory by using the Xen dom0_mem option. For example, if you have 4Gb of memory, give dom0 1Gb in /rpool/boot/grub/menu.lst:

```
...  
kerneld$ /boot/$ISADIR/xen.gz dom0_mem=1024M  
...  
```

If Text-Only Boot Was not Selected in OpenSolaris 2008.11 Installation

When installing OpenSolaris 2008.11, the "text-only" boot option from the GRUB menu should be selected. This is the second entry. However, the following workaround can be used if the text-only boot option is not selected.
1. When the GNU GRand Unified Bootloader (GRUB) boot menu appears on the console, type e for edit before the boot selection times out and starts booting the default entry.

2. The edit menu screen that pops up will have several lines on it. Use the down arrow key to scroll down to the line foreground, and type d for delete.

3. Repeat the Step 2 process for the line background. Use the down arrow key to scroll down to the line that says background, and type d for delete.

4. Using the arrow keys, scroll to the kernel line that contains the boot arguments, and type e for edit. Then, delete the entry for console=graphics.

5. Press Return to exit line edit and save the changes, and then type b to continue the boot.

---

### Verifying System Elements

#### How to Verify Devices in /dev/xen

- Change directories to /dev/xen and list the contents:

  ```
  # cd /dev/xen
  # ls
  balloon domcaps evtchn privcmd xenbus
  ```

#### How to Verify That the xvm Hypervisor Services Are Started

1. Become superuser, or assume the appropriate role.

2. Verify that the xvm services are running.

   ```
   # svcs -a | grep xvm
   ```

   If the system displays the following, the services are not running:

   ```
   disabled Dec_07 svc:/system/xvm/store:default
   disabled Dec_07 svc:/system/xvm/xend:default
   disabled Dec_07 svc:/system/xvm/console:default
   disabled Dec_07 svc:/system/xvm/domains:default
   ```
3 If the services are not running, verify that you booted an `i86xpv` kernel.
   # uname -i
   i86xpv
   Reboot if necessary.

4 If the correct kernel is running, enable the services.
   # svcadm enable xvm/store
   # svcadm enable xvm/xend
   # svcadm enable xvm/console
   # svcadm enable xvm/domains
   You are now ready to create guest domains (domUs).

### Tracing the Interaction Between dom0 and the Hypervisor

Due to the isolation of the hypervisor from dom0, there is currently no way to apply DTrace directly to the hypervisor. There is an `xpv` DTrace provider that allows you to trace the interaction between dom0 and the hypervisor.

To list the available DTrace probes, use the following command line:

```bash
# dtrace -l -i 'xpv:::'
```

To enable all probes, use the following command line

```bash
# dtrace -n 'xpv::: {}'
```

### Guest Issues

#### Known OpenSolaris 2009.06 Guest Issues

- If you are running an OpenSolaris 2009.06 HVM guest on Hyper-V SP2 (RC build), you'll need to update your OpenSolaris 2009.06 to at least `snv_113` to get the fix for CR 6768204, “dnet interface takes a long time to resume after plumb/unplumb in Hyper-V virtual machine.” Without the update, the network interface doesn’t work for several minutes after the first interface unplumb/plumb operation. Obtain the update from the OpenSolaris Packaging Repository (http://pkg.opensolaris.org/dev).

- If you're running an OpenSolaris 2009.06 HVM guest on OpenSolaris 2009.06 dom0, you'll need to apply the workaround for CR 6828707 for XServer to start on the guest.
The issue is how long gdm should wait before it assumes a started Xserver is defunct. The default set in /usr/share/gdm/factory-defaults.conf is:

GdmXserverTimeout=10

The default should be increased. To override the default and set the timeout to the recommended value, add the following to /etc/X11/gdm/custom.conf:

GdmXserverTimeout=30

If the workaround is not used, the user will not see the GUI login window. If booted in text mode, only a console login window will be available. If booted in graphics mode, only the OpenSolaris startup screen will be seen.

Known Issues With Running Solaris 10 Guests on Other Operating Systems

- CR 6810371 HCTS CPU test fail with "can not fork: too many processes" if the virtual machine uses a dynamic hard disk.
  This problem affects Solaris 10 5/09 HVM guests. The workaround is to either change hard disk of the virtual machine to fixed-size, or to create a big file and point system swap to use that file.

For More Information

For additional documentation and more information about the hypervisor feature, visit the OpenSolaris community (http://www.opensolaris.org/os/community/xen). The FAQ at this site will be updated to contain the latest issues.

For the latest information on dom0, see the document dom0 configuration for admins (http://www.opensolaris.org/os/community/xen/docs/configuring-dom0/). See the document New Network Options, Including Limiting Bandwidth and Setting a VLAN ID, for Virtual Network Interfaces Attached to a Guest at this site to learn about new conventions for network options in virt-install.

xVM 3.3 documentation can be found on the OpenSolaris Community: Xen site.
## Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AMD-V</strong></td>
<td>AMD’s extensions that make the x86 architecture HVM-capable.</td>
</tr>
<tr>
<td><strong>backend driver</strong></td>
<td>Half of a virtual driver, providing an interface between the virtual device and an underlying real device. See frontend driver.</td>
</tr>
<tr>
<td><strong>bare metal environment</strong></td>
<td>A virtual environment where the virtualization product is directly installed on physical hardware, acting like a host operating system. The opposite of a hosted environment.</td>
</tr>
<tr>
<td><strong>blessed</strong></td>
<td>In Perl, the term used to denote class membership of an object.</td>
</tr>
<tr>
<td><strong>brand</strong></td>
<td>An instance of the BrandZ functionality, which provides non-global zones that contain non-native operating environments used for running applications.</td>
</tr>
<tr>
<td><strong>branded zone</strong></td>
<td>An isolated environment in which to run non-native applications in non-global zones.</td>
</tr>
<tr>
<td><strong>cap</strong></td>
<td>A limit that is placed on system resource usage.</td>
</tr>
<tr>
<td><strong>capping</strong></td>
<td>The process of placing a limit on system resource usage.</td>
</tr>
<tr>
<td><strong>control domain</strong></td>
<td>Fully privileged control domain 0, can create and destroy other domains, access real hardware, and so forth. See also guest domain.</td>
</tr>
<tr>
<td><strong>default pool</strong></td>
<td>The pool created by the system when pools are enabled. See also resource pool.</td>
</tr>
<tr>
<td><strong>default processor set</strong></td>
<td>The processor set created by the system when pools are enabled. See also processor set.</td>
</tr>
<tr>
<td><strong>disjoint</strong></td>
<td>A type of set in which the members of the set do not overlap and are not duplicated.</td>
</tr>
<tr>
<td><strong>domain</strong></td>
<td>Virtual machine instance.</td>
</tr>
<tr>
<td><strong>dynamic configuration</strong></td>
<td>Information about the disposition of resources within the resource pools framework for a given system at a point in time.</td>
</tr>
<tr>
<td><strong>dynamic reconfiguration</strong></td>
<td>On SPARC based systems, the ability to reconfigure hardware while the system is running. Also known as DR.</td>
</tr>
<tr>
<td><strong>extended accounting</strong></td>
<td>A flexible way to record resource consumption on a task basis or process basis in the Solaris Operating System.</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>fair share scheduler</td>
<td>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.</td>
</tr>
<tr>
<td>frontend driver</td>
<td>A virtual device and its associated driver in a guest domain that communicates with a backend hosted in another guest domain. See backend driver.</td>
</tr>
<tr>
<td>FSS</td>
<td>See fair share scheduler.</td>
</tr>
<tr>
<td>global administrator</td>
<td>An administrator with superuser privileges or the Primary Administrator role. When logged in to the global zone, the global administrator can monitor and control the system as a whole. See also zone administrator.</td>
</tr>
<tr>
<td>global scope</td>
<td>Actions that apply to resource control values for every resource control on the system.</td>
</tr>
<tr>
<td>global zone</td>
<td>The zone contained on every 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.</td>
</tr>
<tr>
<td>guest domain</td>
<td>Completely unprivileged virtual machine. Only virtual devices are accessible. Also see control domain.</td>
</tr>
<tr>
<td>heap</td>
<td>Process-allocated scratch memory.</td>
</tr>
<tr>
<td>HVM</td>
<td>Hardware-assisted virtual machine. These are virtual machines that can take advantage of Intel-VT and AMD-V extensions.</td>
</tr>
<tr>
<td>hypervisor</td>
<td>A layer between software environments and physical hardware that virtualizes the system’s hardware.</td>
</tr>
<tr>
<td>Linux branded zone</td>
<td>Non-global zone that provides a Linux environment for applications.</td>
</tr>
<tr>
<td>local scope</td>
<td>Local actions taken on a process that attempts to exceed the control value.</td>
</tr>
<tr>
<td>locked memory</td>
<td>Memory that cannot be paged.</td>
</tr>
<tr>
<td>memory cap enforcement threshold</td>
<td>The percentage of physical memory utilization on the system that will trigger cap enforcement by the resource capping daemon.</td>
</tr>
<tr>
<td>naming service database</td>
<td>In the Projects and Tasks (Overview) chapter of this document, a reference to both LDAP containers and NIS maps.</td>
</tr>
<tr>
<td>non-global zone</td>
<td>A virtualized operating system environment created within a single instance of the Solaris Operating System. The Solaris Zones software partitioning technology is used to virtualize operating system services.</td>
</tr>
<tr>
<td>non-global zone administrator</td>
<td>See zone administrator.</td>
</tr>
<tr>
<td>page in</td>
<td>To read data from a file into physical memory one page at a time.</td>
</tr>
<tr>
<td>page out</td>
<td>To relocate pages to an area outside of physical memory.</td>
</tr>
<tr>
<td>term</td>
<td>definition</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>paravirtualization</td>
<td>The paravirtualized domU operating system is ported to run on top of the hypervisor, and uses virtual network, disk, and console devices.</td>
</tr>
<tr>
<td>pool</td>
<td>See resource pool.</td>
</tr>
<tr>
<td>pool daemon</td>
<td>The <code>poold</code> system daemon that is active when dynamic resource allocation is required.</td>
</tr>
<tr>
<td>processor set</td>
<td>A disjoint grouping of CPUs. Each processor set can contain zero or more processors. A processor set is represented in the resource pools configuration as a resource element. Also referred to as a pset. See also disjoint.</td>
</tr>
<tr>
<td>project</td>
<td>A network-wide administrative identifier for related work.</td>
</tr>
<tr>
<td>resident set size</td>
<td>The size of the resident set. The resident set is the set of pages that are resident in physical memory.</td>
</tr>
<tr>
<td>resource</td>
<td>An aspect of the computing system that can be manipulated with the intent to change application behavior.</td>
</tr>
<tr>
<td>resource capping daemon</td>
<td>A daemon that regulates the consumption of physical memory by processes running in projects that have resource caps defined.</td>
</tr>
<tr>
<td>resource consumer</td>
<td>Fundamentally, a Solaris process. Process model entities such as the project and the task provide ways of discussing resource consumption in terms of aggregated resource consumption.</td>
</tr>
<tr>
<td>resource control</td>
<td>A per-process, per-task, or per-project limit on the consumption of a resource.</td>
</tr>
<tr>
<td>resource management</td>
<td>A functionality that enables you to control how applications use available system resources.</td>
</tr>
<tr>
<td>resource partition</td>
<td>An exclusive subset of a resource. All of the partitions of a resource sum to represent the total amount of the resource available in a single executing Solaris instance.</td>
</tr>
<tr>
<td>resource pool</td>
<td>A configuration mechanism that is used to partition machine resources. A resource pool represents an association between groups of resources that can be partitioned.</td>
</tr>
<tr>
<td>resource set</td>
<td>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.</td>
</tr>
<tr>
<td>RSS</td>
<td>See resident set size.</td>
</tr>
<tr>
<td>scanner</td>
<td>A kernel thread that identifies infrequently used pages. During low memory conditions, the scanner reclaims pages that have not been recently used.</td>
</tr>
<tr>
<td>Solaris Container</td>
<td>A complete runtime environment for applications. Resource management and Solaris Zones software partitioning technology are both parts of the container.</td>
</tr>
<tr>
<td>Solaris Containers for Linux Applications</td>
<td>A technology that enables the creation of a runtime environment for Linux applications in a non-global zone on x86 or x64 machines running the Solaris Operating System.</td>
</tr>
<tr>
<td>Solaris Zones</td>
<td>See Solaris Container. A software partitioning technology used to virtualize operating system services and provide an isolated, secure environment in which to run applications.</td>
</tr>
</tbody>
</table>
### Sparse Root Zone
A type of non-global zone that has `inherit-pkg-dir` resources and optimizes the sharing of objects.

### 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.

### VT-x
Intel's extensions that make the x86 architecture HVM-capable.

### Whole Root Zone
A type of non-global zone that does not have `inherit-pkg-dir` resources.

### Working Set Size
The size of the working set. The working set is the set of pages that the project workload actively uses during its processing cycle.

### Workload
An aggregation of all processes of an application or group of applications.

### WSS
See also working set size.

### Zone Administrator
An administrator having the Zone Management profile. The privileges of a zone administrator are confined to a non-global zone.

See also global administrator.

### Zone State
The status of a non-global zone. The zone state is one of configured, incomplete, installed, ready, running, or shutting down.
Index

A
acctadm command, 77
activating extended accounting, 76-79
administering data-links, 395
administering resource pools, 163
attribute, project.pool, 149

capped-cpu resource, 217, 232
capped-memory, 232
capped-memory resource, 219
changing resource controls temporarily, 96
clones
ZFS, 284, 466-468
cloning a zone, 271, 284
cloning an lx branded zone, 466-468
commands
extended accounting, 71
fair share scheduler (FSS), 118
lx branded zones, 418
projects and tasks, 52
resource controls, 97
zones, 377
configurable privileges, lx branded zone, 428
configurable privileges, zone, 225
configuration, rcapd, 127
configuring branded zones, 439
configuring resource controls, 85
configuring zones, tasks, 241
control domain, 499
CPU share configuration, 114
creating resource pools, 150

capped-cpu, 424

dedicated-cpu resource, 231
default processor set, 144
default project, 46
default resource pool, 144

B
binding to resource pool, 184
boot arguments and zones, 279, 462
/boot/grub/menu.lst, 515
bootargs property, 230
booting a zone, 278
booting an lx branded zone, 461
booting Solaris control domain, 515
brand, 414
branded zone, 202, 414
configuration, 439
device support, 415
file system support, 415
halt, 452
privileges, 416
reboots, 452
running processes, 415
BrandZ, 202, 414

capped-memory, 232

D
default processor set, 144
default project, 46
default resource pool, 144

capped-memory, 232
deleting a zone, 286, 469
DHCP, exclusive-IP zone, 220
disabling dynamic resource pools, 167
disabling resource capping, 137
disabling resource pools, 167
displaying extended accounting status, 77
domain, states, 512
domain management, virsh, 538
domains, viewing, 517
DRP, 145
dtrace_proc, 231, 373, 385
dtrace_user, 231, 373, 385
dynamic pools configuration, 147
dynamic resource pools
  disabling, 167
  enabling, 167

E
enabling dynamic resource pools, 167
enabling resource capping, 136
enabling resource pools, 167
entry format, /etc/project file, 48
/etc/project
  entry format, 48
  file, 47
/etc/user_attr file, 46
exact file, 68
exclusive-IP zone, 220
extended accounting
  activating, 76-79
  chargeback, 68
  commands, 71
  file format, 68
  overview, 67
  SMF, 70
  status, displaying, 77

F
fair share scheduler (FSS), 110, 218, 424
  and processor sets, 115
fair share scheduler (FSS) (Continued)
  configuration, 120
  project.cpu-shares, 110
  share definition, 110
FSS, See fair share scheduler (FSS)

G
global administrator, 204, 206
global zone, 204
guests "known to work" on hypervisor, 497

H
halting a branded zone, 452
  troubleshooting, 452
halting a zone, 269, 281
  troubleshooting, 269
halting an \x branded zone, 463
hardware, supported on hypervisor, 495
host ID in a zone, 315
hostid property in a zone, 315
HVM, 497
guest, 503
HVM + PVIO, 497
hypervisor
  credit scheduler, 501
  "known to work" guests, 497
  supported guests, 497
  supported hardware, 495
hypervisor management, xm, 544
hypervisor virtualization, supported modes, 502

I
implementing resource pools, 149
inherit-pkg-dir resource, native brand, 233
installing a zone, 274, 275
installing an \x branded zone, 456
installing applications in \x branded zone, 486
interactive packages, 322
interprocess communication (IPC), See resource controls
IP Filter, exclusive-IP zone, 220
IP routing, exclusive-IP zone, 220
ip-type property, 231
ipkg, zone, 202
ipkg zone
  manual syncing, 200
  new in 2009.06, 200
IPMP, exclusive-IP zone, 220
IPsec, used in zone, 371

L
libexacct library, 68
limitpriv property, 231
Linux archives, 456
Linux branded zone overview, 416
listing zones, 275, 456
locked memory cap, 219
login, remote zone, 291
\x branded zone
  application support, 417
  boot procedure, 461
capped-cpu, 435
capped-memory, 425, 435
cap, 466-468
commands used with, 418
configurable privileges, 428
configuration overview, 422
configuring, 430
dedicated-cpu, 435
devices, 429
enabling networking, 459
file systems, 429
halting, 463
installation methods, 450
installation overview, 449
installing, 456
installing applications, 486
list, 456
login overview, 471
\x migration trial run, 482
migrate, 478

\x branded zone (Continued)
move, 477-478
overview, 416
passwords, 452
populating, 451
privileges, 429
rebooting, 464
resource type properties, 436
resource types, 433
scheduling-class, 435
Sun package clusters, 450
supported configuration when packages added, 485
supported distributions, 416
supported processor types, 416
uninstalling, 465
upgrading CentOS distribution, 485
upgrading packages, 486
upgrading Red Hat distribution, 485
zone-wide resource controls, 438
\x branded zone installation methods, 450

M
memory cap enforcement threshold, 128
migrating a zone, 306
migrating an \x zone, 478
moving a zone, 305-306
moving an \xzone, 477-478

N
native
  zone, 202, 414
native brand, inherit-pkg-dir resource, 233
net resource
  exclusive-IP zone, 220
  shared-IP zone, 220
networking, exclusive-IP, 362
networking, shared-IP, 360
networking in an \x branded zone, 459
non-global zone, 204
non-global zone administrator, 204
Index

non-native
zone, 202, 414

P
p2v
  flarcreate, 314
  image creation, 314
P2V, system evaluation, 313
p2v image creation, 314
package operations, 322
packages, interactive, 322
PAM (pluggable authentication module), identity
management, 48
paravirtualization
  devices and drivers, 502
  overview, 502
passwords in an lx zone, 452
patches generated for packages, 321
Perl interface, 72
physical memory cap, 219
pluggable authentication module, See PAM
pool property, 230
poold
  asynchronous control violation, 161
  configurable features, 156
  constraints, 152
  control scope, 160
  cpu-pinned property, 153
  description, 151
  dynamic resource allocation, 145
  logging information, 157
  objectives, 153
  synchronous control violation, 161
  pools, 144
poolstat
  description, 162
  output format, 162
  usage examples, 186
populating a zone, 266
populating an lx branded zone, 451
privilege levels, threshold values, 91
privileges in a zone, 367
privileges in an lx branded zone, 429
project
  active state, 111
  definition, 46
  idle state, 111
  with zero shares, 110
project 0, 114
project.cpu-shares, 114
project database, 47
project.pool attribute, 149
project system, See project 0
putacct system call, 69

R
rcap.max-rss attribute, 126
rcapadm command, 127
rcapd
  configuration, 127
  sample intervals, 131
  scan intervals, 130
rcapd daemon, 125
rcapstat command, 131
rctls, 83
  See resource controls
ready a zone, 278
rebooting a branded zone, 452
rebooting a zone, 269, 282
rebooting an lx branded zone, 464
remote zone login, 291
removing resource pools, 184
renaming a zone, 259
resource cap, 125
resource capping
  disabling, 137
  enabling, 136
resource capping daemon, 125
resource controls
  changing temporarily, 96
  configuring, 85
  definition, 83
  global actions, 92
  inf value, 95
  interprocess communication (IPC), 84
  list of, 86

558  System Administration Guide: Virtualization Using the Solaris Operating System • October 2009 (Beta)
resource controls (Continued)
   local actions, 85, 92
   overview, 83
   temporarily updating, 96
   threshold values, 85, 92
   zone-wide, 222, 438
resource limits, 84
resource management
   constraints, 39
   definition, 37
   partitioning, 40
   scheduling, 39
   tasks, 51
resource pools, 144
   activating configuration, 183
   administering, 163
   binding to, 184
   configuration elements, 148
   creating, 150
   disabling, 167
   dynamic reconfiguration, 149
   enabling, 167
   /etc/pooladm.conf, 147
   implementing, 149
   properties, 148
   removing, 184
   removing configuration, 183
   static pools configuration, 147
   rlimits, See resource limits
   running DTrace in a zone, 373, 385
Sun xVM hypervisor
   control domain, 499
   introduction, 493
   operation, 493
SUNW_PKG_ALLZONES package parameter, 332
SUNW_PKG_HOLLOW package parameter, 334
SUNW_PKG_THISZONE package parameter, 335
supported guests on hypervisor, 497
swap space cap, 219
system evaluation for P2V, 313
T
tasks, resource management, 51
temporarily updating resource controls, 96
temporary pool, 217
threshold values, resource controls, 91
U
uninstalling a zone, 283, 465
V
/var/adm/exacct directory, 70
verifying a zone, 274
viewing domains, \, 517
virsh, domain management, 538
W
whole root zone, 201
X
xm, hypervisor management, 544

S
  scheduling-class property, 231
  scheduling classes, 117
  server consolidation, 41
  set zone.cpu-shares in global zone, 260
  setting resource pool attributes, 184
  shared-IP zone, 220
  snapshots
     ZFS, 284, 466-468
Solaris control domain, booting, 515
  sparse root zone, 201
  native brand, 201
Z

ZFS
- clones, 284, 466-468
- dataset, 232
- snapshots, 284, 466-468

zone
- adding packages, 324
- adding patches, 336
- administering data-links, 395
- boot arguments, 269, 279, 462
- boot single-user, 280, 462
- bootargs property, 230
- booting, 278
- branded, 414
- capped-memory, 219, 232
- characteristics by type, 205
- clone, 271, 284
- commands used in, 377
- configurable privileges, 225
- configuring, 225
- creating, 206
- dataset, 232
- dedicated-cpu, 231, 232
- definition, 201
- delete, 286, 469
- disk space, 244
- exclusive-IP, 220
- features, 210
- halting, 269, 281
- installing, 275
- interactive mode, 292
- internal configuration, 290
- ip-type, 231
- ipkg, 202
- IPsec, 371
- limitpriv, 231
- list, 275
- login overview, 289
- migrate, 306
- migrating from unusable machine, 312
- migration trial run, 311
- move, 305-306
- native, 202, 414
- network address, 246

zone (Continued)
- networking, exclusive-IP, 362
- networking, shared-IP, 360
- non-interactive mode, 292
- non-native, 202, 414
- package and patch overview, 320
- package rules, 322
- PatchPro support, 339
- pool, 230
- populating, 266
- privileges, 367
- ready state, 278
- rebooting, 269, 282
- removing packages, 327
- removing patches, 339
- rename, 259
- resource controls, 222, 438
- resource type properties, 234
- resource types, 229
- running DTrace in, 373
- scheduling-class, 231
- scope, 320
- shared-IP, 220
- state model, 207
- states, 207
- uninstalling, 283
- upgrade on attach, 307
- UUID, 276, 459
- verify, 274
- zone, zone-wide resource controls, 229
- zone administrator, 206
- zone configuration
  - overview, 216
  - script, 255, 445
  - tasks, 241
- zone console login, console login mode, 290
- zone.cpu-cap resource control, 222
- zone.cpu-shares resource control, 223
- zone host name, 246
- zone ID, 204
- zone installation
  - overview, 265
  - tasks, 274
zone login
   failsafe mode, 291
   remote, 291
Zone Management profile, 398
zone.max-locked-memory resource control, 223
zone.max-wps resource control, 223
zone.max-msg-1ds resource control, 223
zone.max-sem-1ds resource control, 223
zone.max-shm-1ds resource control, 223
zone.max-shm-memory resource control, 223
zone.max-swap resource control, 223
zone migration trial run, 311, 482
zone name, 204
zone node name, 353
zone root file system models, 201
zone size
   restricting, 245, 421
zone—wide resource controls, 222
zone—wide resource controls, 433
zoneadm
   mark subcommand, 277, 460
zoneadm command, 265
zoneadmd daemon, 268
zonecfg
   capped-cpu, 217, 424
   entities, 229, 433
   global zone, 249
   in global zone, 225
   \x branded zone process, 422
   modes, 226, 430
   operations, 216
   procedure, 249, 440
   scope, 226, 430
   scope, global, 226, 430
   scope, resource specific, 226, 430
   subcommands, 226, 431
   temporary pool, 217
zonecfg command, 249
zonepath
   automatically created if on ZFS, 275, 456
   prevent automatic creation if on ZFS, 275, 456
zones commands, 377
zsched process, 268