## Contents

Preface ................................................................................................................................................... 19

Part I  Resource Management ............................................................................................................ 23

1  Introduction to Solaris 10 Resource Manager .............................................................................. 25
   Resource Management Overview ................................................................. 25
   Resource Classifications .................................................................................. 26
   Resource Management Control Mechanisms .................................................. 27
   Resource Management Configuration .............................................................. 28
   Interaction With Solaris Zones ........................................................................... 28
   When to Use Resource Management .................................................................. 28
   Server Consolidation ........................................................................................... 28
   Supporting a Large or Varied User Population .................................................. 29
   Setting Up Resource Management (Task Map) .................................................. 30

2  Projects and Tasks (Overview) ....................................................................................................... 33
   What’s New in Project Database and Resource Control Commands for Solaris 10? ........ 33
   Project and Task Facilities .................................................................................. 34
   Project Identifiers ............................................................................................... 34
   Determining a User’s Default Project ............................................................... 35
   Setting User Attributes With the useradd, usermod, and passwd Commands ........ 35
   project Database ............................................................................................... 36
   PAM Subsystem ................................................................................................. 36
   Naming Services Configuration .......................................................................... 37
   Local /etc/project File Format ......................................................................... 37
   Project Configuration for NIS ............................................................................. 39
   Project Configuration for LDAP ........................................................................ 39
3 Administering Projects and Tasks ................................................................. 43
   Administering Projects and Tasks (Task Map) .............................................. 43
   Example Commands and Command Options .............................................. 44
      Command Options Used With Projects and Tasks ................................. 44
      Using cr on and su With Projects and Tasks ........................................... 46
   Administering Projects ............................................................................... 47
      ▼ How to Define a Project and View the Current Project ......................... 47
      ▼ How to Delete a Project From the /etc/project File ......................... 50
         How to Validate the Contents of the /etc/project File ...................... 51
         How to Obtain Project Membership Information ................................ 51
      ▼ How to Create a New Task ................................................................. 51
      ▼ How to Move a Running Process Into a New Task .............................. 52
   Editing and Validating Project Attributes .................................................. 52
      ▼ How to Add Attributes and Attribute Values to Projects .................. 53
      ▼ How to Remove Attribute Values From Projects ............................... 53
      ▼ How to Remove a Resource Control Attribute From a Project .......... 54
      ▼ How to Substitute Attributes and Attribute Values for Projects ......... 54
      ▼ How to Remove the Existing Values for a Resource Control Attribute .. 55

4 Extended Accounting (Overview) ................................................................. 57
   What’s New in Extended Accounting for Solaris 10? .................................. 57
   Introduction to Extended Accounting ......................................................... 58
   How Extended Accounting Works ............................................................. 58
      Extensible Format .................................................................................... 59
      exact Records and Format ...................................................................... 59
      Using Extended Accounting on a Solaris System with Zones Installed .... 60
   Extended Accounting Configuration ......................................................... 60
   Commands Used With Extended Accounting ............................................ 60
   Perl Interface to libexacct ......................................................................... 61
## 5 Administering Extended Accounting (Tasks)

Administrering the Extended Accounting Facility (Task Map) ................................................................. 65
Using Extended Accounting Functionality ........................................................................................................ 65

- How to Activate Extended Accounting for Processes, Tasks, and Flows ........................................ 66
  - How to Activate Extended Accounting With a Startup Script ......................................................... 66
  - How to Display Extended Accounting Status ................................................................................ 67
  - How to View Available Accounting Resources .............................................................................. 67

- How to Deactivate Process, Task, and Flow Accounting ........................................................................ 68

Using the Perl Interface to libexacct ........................................................................................................ 68
  - How to Recursively Print the Contents of an exacct Object ......................................................... 68
  - How to Create a New Group Record and Write It to a File ......................................................... 70
  - How to Print the Contents of an exacct File .................................................................................. 71

Example Output From Sun::Solaris::Exacct::Object->dump() ................................................................. 72

## 6 Resource Controls (Overview)

What's New in Resource Controls for Solaris 10? ......................................................................................... 73
Resource Controls Concepts ......................................................................................................................... 74
  - Resource Limits and Resource Controls ....................................................................................... 74
  - Interprocess Communication and Resource Controls .................................................................... 75
  - Resource Control Constraint Mechanisms .................................................................................... 75
  - Project Attribute Mechanisms ...................................................................................................... 75

Configuring Resource Controls and Attributes ......................................................................................... 76
  - Available Resource Controls ........................................................................................................ 76
  - Units Support ............................................................................................................................... 79
  - Resource Control Values and Privilege Levels ............................................................................. 80
  - Global and Local Actions on Resource Control Values ................................................................. 81
  - Resource Control Flags and Properties ......................................................................................... 83
  - Resource Control Enforcement .................................................................................................... 84

Global Monitoring of Resource Control Events ....................................................................................... 84
Applying Resource Controls ....................................................................................................................... 85
  - Temporarily Updating Resource Control Values on a Running System ....................................... 85
    - Updating Logging Status ........................................................................................................... 85
    - Updating Resource Controls .................................................................................................. 85
  - Commands Used With Resource Controls ................................................................................. 86
7 Administering Resource Controls (Tasks) ................................................................. 87
  Administering Resource Controls (Task Map) .......................................................... 87
  Setting Resource Controls ......................................................................................... 88
    ▼ How to Set the Maximum Number of LWPs for Each Task in a Project ............... 88
    ▼ How to Set Multiple Controls on a Project ......................................................... 89
  Using the prctl Command .......................................................................................... 91
    ▼ How to Use the prctl Command to Display Default Resource Control Values ...... 91
    ▼ How to Use the prctl Command to Display Information for a Given Resource Control . 93
    ▼ How to Use prctl to Temporarily Change a Value ............................................... 93
    ▼ How to Use prctl to Lower a Resource Control Value ....................................... 94
    ▼ How to Use prctl to Display, Replace, and Verify the Value of a Control on a Project .... 94
  Using rctladm ............................................................................................................. 95
    How to Use rctladm .................................................................................................. 95
  Using ipc ..................................................................................................................... 96
    How to Use ipc .......................................................................................................... 96
  Capacity Warnings ...................................................................................................... 96
    ▼ How to Determine Whether a Web Server Is Allocated Enough CPU Capacity .......... 97

8 Fair Share Scheduler (Overview) ............................................................................ 99
  Introduction to the Scheduler .................................................................................... 99
  CPU Share Definition ............................................................................................... 100
  CPU Shares and Process State .................................................................................. 101
  CPU Share Versus Utilization ................................................................................... 101
  CPU Share Examples ............................................................................................... 101
    Example 1: Two CPU-Bound Processes in Each Project ....................................... 102
    Example 2: No Competition Between Projects ..................................................... 102
    Example 3: One Project Unable to Run ................................................................... 103
  FSS Configuration ...................................................................................................... 103
    Projects and Users ................................................................................................. 103
    CPU Shares Configuration ..................................................................................... 104
  FSS and Processor Sets ............................................................................................ 105
    FSS and Processor Sets Examples ......................................................................... 106
  Combining FSS With Other Scheduling Classes ....................................................... 107
  Setting the Scheduling Class for the System ............................................................. 108
  Scheduling Class on a System with Zones Installed .................................................. 108
Commands Used With FSS ................................................................. 109

9 Administering the Fair Share Scheduler (Tasks) ......................................................... 111
  Administering the Fair Share Scheduler (Task Map) ................................................. 111
  Monitoring the FSS ......................................................................................... 112
    ▼ How to Monitor System CPU Usage by Projects ........................................... 112
    ▼ How to Monitor CPU Usage by Projects in Processor Sets ......................... 112
  Configuring the FSS ....................................................................................... 112
    ▼ How to Make FSS the Default Scheduler Class ........................................... 113
    ▼ How to Manually Move Processes From the TS Class Into the FSS Class ........ 113
    ▼ How to Manually Move Processes From All User Classes Into the FSS Class .... 114
    ▼ How to Manually Move a Project’s Processes Into the FSS Class ................ 114
    How to Tune Scheduler Parameters ............................................................. 115

10 Physical Memory Control Using the Resource Capping Daemon (Overview) ............ 117
  What’s New in Physical Memory Control Using the Resource Capping Daemon? .... 117
  Introduction to the Resource Capping Daemon .................................................... 118
  How Resource Capping Works .......................................................................... 118
  Attribute to Limit Physical Memory Usage .......................................................... 119
  rcapd Configuration ......................................................................................... 119
    Using the Resource Capping Daemon on a System With Zones Installed .............. 120
    Memory Cap Enforcement Threshold ............................................................. 120
    Determining Cap Values ................................................................................. 121
    rcapd Operation Intervals ................................................................................ 122
  Monitoring Resource Utilization With rcapstat ................................................... 123
  Commands Used With rcapd .............................................................................. 124

11 Administering the Resource Capping Daemon (Tasks) ............................................. 127
  Configuring and Using the Resource Capping Daemon (Task Map) ......................... 127
    ▼ How to Set the Memory Cap Enforcement Threshold ................................... 128
    ▼ How to Set Operation Intervals ................................................................... 129
    ▼ How to Enable Resource Capping ............................................................... 129
    ▼ How to Disable Resource Capping .............................................................. 130
Producing Reports With rcapstat ................................................................. 130
Reporting Cap and Project Information .................................................. 130
Monitoring the RSS of a Project .............................................................. 131
Determining the Working Set Size of a Project ....................................... 132
Reporting Memory Utilization and the Memory Cap Enforcement Threshold ............ 133

12 Resource Pools (Overview) ................................................................. 135
Introduction to Resource Pools ................................................................. 136
Introduction to Dynamic Resource Pools ................................................ 137
About Enabling and Disabling Resource Pools and Dynamic Resource Pools .......... 137
Resource Pools Used in Zones ................................................................. 137
When to Use Pools .................................................................................. 138
Resource Pools Framework ....................................................................... 139
Implementing Pools on a System .............................................................. 141
project pool Attribute ............................................................................. 141
SPARC: Dynamic Reconfiguration Operations and Resource Pools ............ 142
Creating Pools Configurations ................................................................. 142
Directly Manipulating the Dynamic Configuration ...................................... 143
poold Overview ....................................................................................... 143
Managing Dynamic Resource Pools ......................................................... 144
Configuration Constraints and Objectives .............................................. 144
Configuration Constraints ....................................................................... 144
Configuration Objectives ........................................................................ 145
poold Properties ..................................................................................... 148
poold Features That Can Be Configured .................................................. 149
poold Monitoring Interval ...................................................................... 149
poold Logging Information .................................................................... 149
Logging Location ..................................................................................... 151
Log Management With logadm ............................................................... 151
How Dynamic Resource Allocation Works ............................................ 151
About Available Resources .................................................................... 151
Determining Available Resources .......................................................... 152
Identifying a Resource Shortage .............................................................. 152
Determining Resource Utilization ............................................................ 153
13 Creating and Administering Resource Pools (Tasks) ......................................................... 157
Administering Dynamic Resource Pools (Task Map) ......................................................... 157
Enabling and Disabling the Pools Facility ............................................................... 159
▼ Solaris 10 11/06: How to Enable the Resource Pools Service Using svcadm .......... 159
▼ Solaris 10 11/06: How to Disable the Resource Pools Service Using svcadm .......... 159
▼ Solaris 10 11/06: How to Enable the Dynamic Resource Pools Service Using svcadm .... 160
▼ Solaris 10 11/06: How to Disable the Dynamic Resource Pools Service Using svcadm .... 162
▼ How to Enable Resource Pools Using pooladm ....................................................... 163
▼ How to Disable Resource Pools Using pooladm ....................................................... 163
Configuring Pools ........................................................................................................ 163
▼ How to Create a Static Configuration .............................................................. 163
▼ How to Modify a Configuration ....................................................................... 165
▼ How to Associate a Pool With a Scheduling Class .............................................. 167
▼ How to Set Configuration Constraints .............................................................. 169
▼ How to Define Configuration Objectives ............................................................ 170
▼ How to Set the poold Logging Level ................................................................. 172
▼ How to Use Command Files With poolcfg ...................................................... 173
Transferring Resources .............................................................................................. 174
▼ How to Move CPUs Between Processor Sets ..................................................... 174
Activating and Removing Pool Configurations ....................................................... 174
▼ How to Activate a Pools Configuration ........................................................... 175
▼ How to Validate a Configuration Before Committing the Configuration .......... 175
▼ How to Remove a Pools Configuration ............................................................ 175
Setting Pool Attributes and Binding to a Pool ....................................................... 176
▼ How to Bind Processes to a Pool ................................................................. 176
▼ How to Bind Tasks or Projects to a Pool ........................................................... 177
▼ How to Set the project.pool Attribute for a Project ........................................ 177
▼ How to Use project Attributes to Bind a Process to a Different Pool ............... 177
### Contents

Using `poolstat` to Report Statistics for Pool-Related Resources .................................................. 178  
Displaying Default `poolstat` Output .......................................................................................... 178  
Producing Multiple Reports at Specific Intervals ..................................................................... 179  
Reporting Resource Set Statistics .......................................................................................... 179

<table>
<thead>
<tr>
<th>14 Resource Management Configuration Example</th>
<th>181</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration to Be Consolidated</td>
<td>181</td>
</tr>
<tr>
<td>Consolidation Configuration</td>
<td>182</td>
</tr>
<tr>
<td>Creating the Configuration</td>
<td>182</td>
</tr>
<tr>
<td>Viewing the Configuration</td>
<td>183</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15 Resource Control Functionality in the Solaris Management Console</th>
<th>189</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the Console (Task Map)</td>
<td>190</td>
</tr>
<tr>
<td>Console Overview</td>
<td>190</td>
</tr>
<tr>
<td>Management Scope</td>
<td>190</td>
</tr>
<tr>
<td>Performance Tool</td>
<td>191</td>
</tr>
<tr>
<td>▼ How to Access the Performance Tool</td>
<td>191</td>
</tr>
<tr>
<td>Monitoring by System</td>
<td>192</td>
</tr>
<tr>
<td>Monitoring by Project or User Name</td>
<td>192</td>
</tr>
<tr>
<td>Resource Controls Tab</td>
<td>194</td>
</tr>
<tr>
<td>▼ How to Access the Resource Controls Tab</td>
<td>195</td>
</tr>
<tr>
<td>Resource Controls You Can Set</td>
<td>196</td>
</tr>
<tr>
<td>Setting Values</td>
<td>197</td>
</tr>
<tr>
<td>Console References</td>
<td>197</td>
</tr>
</tbody>
</table>

| Part II Zones | 199 |

<table>
<thead>
<tr>
<th>16 Introduction to Solaris Zones</th>
<th>201</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zones Overview</td>
<td>201</td>
</tr>
<tr>
<td>When to Use Zones</td>
<td>202</td>
</tr>
<tr>
<td>How Zones Work</td>
<td>204</td>
</tr>
<tr>
<td>Summary of Zone Features</td>
<td>205</td>
</tr>
<tr>
<td>How Non-Global Zones Are Administered</td>
<td>206</td>
</tr>
<tr>
<td>How Non-Global Zones Are Created</td>
<td>206</td>
</tr>
</tbody>
</table>
17 Non-Global Zone Configuration (Overview) ................................................................. 213
What’s New in This Chapter? ......................................................................................... 213
Pre-Installation Configuration Process ........................................................................ 214
Zone Components ........................................................................................................ 214
Zone Name and Path ...................................................................................................... 214
Resource Pool Association ........................................................................................... 214
Zone Interfaces .............................................................................................................. 214
File Systems Mounted in Zones .................................................................................... 214
Configured Devices in Zones ....................................................................................... 215
Zone-Wide Resource Controls ..................................................................................... 215
Configurable Privileges ............................................................................................... 215
Including a Comment for a Zone ................................................................................ 216
Using the zonecfg Command ....................................................................................... 216
zonecfg Modes .............................................................................................................. 217
zonecfg Interactive Mode ............................................................................................. 217
zonecfg Command-File Mode ..................................................................................... 219
Zone Configuration Data .............................................................................................. 219
Resource and Property Types ...................................................................................... 219
Resource Type Properties ............................................................................................ 223
Tecla Command-Line Editing Library ......................................................................... 225

18 Planning and Configuring Non-Global Zones (Tasks) ............................................ 227
Planning and Configuring a Non-Global Zone (Task Map) ........................................... 227
Evaluating the Current System Setup .......................................................................... 228
Disk Space Requirements ............................................................................................ 228
Restricting Zone Size ................................................................................................... 230
Determine the Zone Host Name and Obtain the Network Address ............................. 230
Zone Host Name ........................................................................................................... 230
Zone Network Address ............................................................................................... 231
19 About Installing, Halting, and Uninstalling Non-Global Zones (Overview) ........................................... 247
Zone Installation Concepts .......................................................................................................................... 247
Zone Construction ..................................................................................................................................... 248
The zoneadm Daemon .............................................................................................................................. 249
The zsched Zone Scheduler ....................................................................................................................... 250
Zone Application Environment ................................................................................................................ 250
About Halting, Rebooting, and Uninstalling Zones ............................................................................... 250
Halting a Zone ......................................................................................................................................... 250
Rebooting a Zone ..................................................................................................................................... 251
Zone autoboot ......................................................................................................................................... 251
Uninstalling a Zone ................................................................................................................................. 251
About Cloning Non-Global Zones .......................................................................................................... 251

20 Installing, Booting, Halting, and Uninstalling Non-Global Zones (Tasks) ............................................. 253
Zone Installation (Task Map) .................................................................................................................... 253
Installing and Booting Zones .................................................................................................................. 254
▼ (Optional) How to Verify a Configured Zone Before It Is Installed .................................................... 254
▼ How to Install a Configured Zone ...................................................................................................... 255
▼ (Optional) How to Transition the Installed Zone to the Ready State .................................................. 256
▼ How to Boot a Zone ............................................................................................................................... 256
How to Boot a Zone in Single-User Mode ................................................................. 258
Where to Go From Here ............................................................................................ 258
Halting, Rebooting, Uninstalling, Cloning, and Deleting Non-Global Zones (Task Map) ................................................................. 258
Halting, Rebooting, and Uninstalling Zones ............................................................ 259

- How to Halt a Zone .................................................................................................. 259
- How to Reboot a Zone ............................................................................................ 260
- How to Uninstall a Zone ........................................................................................ 261

Cloning a Non-Global Zone on the Same System ..................................................... 262

- How to Clone a Zone ................................................................................................ 262

Deleting a Non-Global Zone from the System ......................................................... 263

- How to Remove a Non-Global Zone ................................................................. 263

21 Non-Global Zone Login (Overview) ........................................................................ 265
zlogin Command ........................................................................................................ 265
Internal Zone Configuration ....................................................................................... 266
Non-Global Zone Login Methods ............................................................................. 266
   Zone Console Login ............................................................................................... 266
   User Login Methods ............................................................................................. 267
Failsafe Mode ............................................................................................................ 267
Remote Login ............................................................................................................. 267
Interactive and Non-Interactive Modes ................................................................. 267
   Interactive Mode .................................................................................................. 267
   Non-Interactive Mode ......................................................................................... 268

22 Logging In to Non-Global Zones (Tasks) ............................................................... 269
Initial Zone Boot and Zone Login Procedures (Task Map) .......................................... 269
Performing the Initial Internal Zone Configuration .................................................. 270
   How to Log In to the Zone Console to Perform the Internal Zone Configuration .... 270
   How to Use an /etc/sysidcfg File to Perform the Initial Zone Configuration ........ 272
Logging In to a Zone .................................................................................................. 273
   How to Log In to the Zone Console .................................................................... 273
   How to Use Interactive Mode to Access a Zone ................................................... 274
   How to Use Non-Interactive Mode to Access a Zone ........................................... 275
   How to Exit a Non-Global Zone ......................................................................... 275
   How to Use Failsafe Mode to Enter a Zone ........................................................... 275
23 Moving and Migrating Non-Global Zones (Tasks) ................................................................. 279
   Solaris 11/06: Moving a Non-Global Zone ................................................................. 279
   ▼ How to Move a Zone ................................................................................................. 279
   Solaris 11/06: Migrating a Non-Global Zone to a Different Machine ....................... 280
      About Migrating a Zone ......................................................................................... 280
   ▼ How to Migrate A Non-Global Zone ...................................................................... 281
   ▼ How to Move the zonepath to a new Host .............................................................. 282

24 About Packages and Patches on a Solaris System With Zones Installed (Overview) .......... 285
   What's New in Packaging and Patching When Zones Are Installed ......................... 286
   Packaging and Patch Tools Overview ......................................................................... 286
   About Packages and Zones ......................................................................................... 287
      Patches Generated for Packages ........................................................................... 288
      Interactive Packages .............................................................................................. 288
   Keeping Zones in Sync .............................................................................................. 288
      Package Operations Possible in the Global Zone .................................................. 288
      Package Operations Possible in a Non-Global Zone ............................................. 289
   About Adding Packages in Zones .............................................................................. 289
      Using pkgadd in the Global Zone .......................................................................... 289
      Using pkgadd in a Non-Global Zone ...................................................................... 291
   About Removing Packages in Zones ........................................................................... 292
      Using pkgrm in the Global Zone ............................................................................ 292
      Using pkgrm in a Non-Global Zone ....................................................................... 293
   Package Parameter Information ................................................................................... 294
      Setting Package Parameters for Zones .................................................................. 294
      SUNW_PKG_ALLZONES Package Parameter ......................................................... 297
      SUNW_PKG_HOLLOW Package Parameter ............................................................ 299
      SUNW_PKG_THISZONE Package Parameter .......................................................... 300
   Package Information Query ........................................................................................ 301
About Adding Patches in Zones ...................................................................................................... 301
Applying Patches on a Solaris System With Zones Installed ........................................................302
Using patchadd in the Global Zone .............................................................................................. 302
Using patchadd in a Non-Global Zone .......................................................................................... 303
Removing Patches on a Solaris System With Zones Installed ......................................................303
Using patchrm in the Global Zone .............................................................................................. 303
Using patchrm in a Non-Global Zone .......................................................................................... 303
PatchPro Support ..............................................................................................................................304
Product Database ..............................................................................................................................304

25 Adding and Removing Packages and Patches on a Solaris System With Zones Installed
(Tasks) .................................................................................................................................................305
Adding and Removing Packages and Patches on a Solaris System With Zones Installed (Task
Map) ....................................................................................................................................................305
Adding a Package on a Solaris System With Zones Installed .......................................................306
▼ How to Add a Package to the Global Zone Only ....................................................................306
▼ How to Add a Package to the Global Zone and All Non-Global Zones ...............................307
▼ How to Add a Package That Is Installed in the Global Zone to All Non-Global Zones ....308
▼ How to Add a Package to a Specified Non-Global Zone Only ..............................................308
Checking Package Information on a Solaris System With Zones Installed ..............................309
▼ How to Check Package Information in the Global Zone Only .............................................309
▼ How to Check Package Information in a Specified Non-Global Zone Only .......................309
Removing a Package From a Solaris System With Zones Installed ............................................310
▼ How to Remove a Package From the Global Zone and All Non-Global Zones ..............310
▼ How to Remove a Package From a Specified Non-Global Zone Only ..............................310
Applying a Patch to a Solaris System With Zones Installed .......................................................311
▼ How to Apply a Patch to the Global Zone Only ....................................................................311
▼ How to Apply a Patch to the Global Zone and All Non-Global Zones .............................311
▼ How to Apply a Patch to a Specified Non-Global Zone Only ..............................................312
Removing a Patch on a System with Zones Installed ...............................................................312
▼ How to Remove a Patch From the Global Zone and All Non-Global Zones .................312
▼ How to Remove a Patch From a Specified Non-Global Zone Only ..............................312
Checking Package Parameter Settings on a System with Zones Installed ...........................313
▼ (Optional) How to Check the Setting of a Package Already Installed on the System ....313
▼ (Optional) How to Check the Setting of a Package in Software on a CD-ROM ............313
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Solaris Zones Administration (Overview)</td>
<td>315</td>
</tr>
<tr>
<td></td>
<td>What’s New in This Chapter?</td>
<td>315</td>
</tr>
<tr>
<td></td>
<td>Global Zone Visibility and Access</td>
<td>316</td>
</tr>
<tr>
<td></td>
<td>Process ID Visibility in Zones</td>
<td>316</td>
</tr>
<tr>
<td></td>
<td>System Observability in Zones</td>
<td>317</td>
</tr>
<tr>
<td></td>
<td>Non-Global Zone Node Name</td>
<td>317</td>
</tr>
<tr>
<td></td>
<td>File Systems and Non-Global Zones</td>
<td>317</td>
</tr>
<tr>
<td></td>
<td>The -o nosuid Option</td>
<td>317</td>
</tr>
<tr>
<td></td>
<td>Mounting File Systems in Zones</td>
<td>318</td>
</tr>
<tr>
<td></td>
<td>Unmounting File Systems in Zones</td>
<td>320</td>
</tr>
<tr>
<td></td>
<td>Security Restrictions and File System Behavior</td>
<td>320</td>
</tr>
<tr>
<td></td>
<td>Non-Global Zones as NFS Clients</td>
<td>322</td>
</tr>
<tr>
<td></td>
<td>Use of mknod Prohibited in a Zone</td>
<td>322</td>
</tr>
<tr>
<td></td>
<td>Traversing File Systems</td>
<td>323</td>
</tr>
<tr>
<td></td>
<td>Restriction on Accessing A Non-Global Zone From the Global Zone</td>
<td>323</td>
</tr>
<tr>
<td></td>
<td>Networking in Non-Global Zones</td>
<td>324</td>
</tr>
<tr>
<td></td>
<td>Zone Partitioning</td>
<td>324</td>
</tr>
<tr>
<td></td>
<td>Network Interfaces</td>
<td>325</td>
</tr>
<tr>
<td></td>
<td>IP Traffic Between Zones on the Same Machine</td>
<td>325</td>
</tr>
<tr>
<td></td>
<td>IP Network Multipathing in Zones</td>
<td>326</td>
</tr>
<tr>
<td></td>
<td>Device Use in Non-Global Zones</td>
<td>326</td>
</tr>
<tr>
<td></td>
<td>/dev and the /devices Namespace</td>
<td>326</td>
</tr>
<tr>
<td></td>
<td>Exclusive-Use Devices</td>
<td>327</td>
</tr>
<tr>
<td></td>
<td>Device Driver Administration</td>
<td>327</td>
</tr>
<tr>
<td></td>
<td>Utilities That Do Not Work or Are Modified in Non-Global Zones</td>
<td>328</td>
</tr>
<tr>
<td></td>
<td>Running Applications in Non-Global Zones</td>
<td>328</td>
</tr>
<tr>
<td></td>
<td>Resource Controls Used in Non-Global Zones</td>
<td>328</td>
</tr>
<tr>
<td></td>
<td>Fair Share Scheduler on a Solaris System With Zones Installed</td>
<td>329</td>
</tr>
<tr>
<td></td>
<td>FSS Share Division in a Non-Global Zone</td>
<td>329</td>
</tr>
<tr>
<td></td>
<td>Share Balance Between Zones</td>
<td>329</td>
</tr>
<tr>
<td></td>
<td>Extended Accounting on a Solaris System With Zones Installed</td>
<td>330</td>
</tr>
<tr>
<td></td>
<td>Privileges in a Non-Global Zone</td>
<td>330</td>
</tr>
<tr>
<td></td>
<td>Using IP Security Architecture in Zones</td>
<td>333</td>
</tr>
<tr>
<td></td>
<td>Using Solaris Auditing in Zones</td>
<td>334</td>
</tr>
<tr>
<td></td>
<td>Configuring Audit in the Global Zone</td>
<td>334</td>
</tr>
<tr>
<td></td>
<td>Configuring User Audit Characteristics in a Non-Global Zone</td>
<td>335</td>
</tr>
</tbody>
</table>
Contents

▼ How to Change the zone.cpu-shares Value in a Zone Dynamically ......................... 360
Using Rights Profiles in Zone Administration .......................................................... 360
▼ How to Assign the Zone Management Profile ...................................................... 360
   Example—Using Profile Shells With Zone Commands .......................................... 361
Back up a Solaris System With Installed Zones ....................................................... 361
▼ How to Use ufsdump to Perform Backups .............................................................. 361
▼ How to Create a UFS Snapshot Using fssnap ....................................................... 362
▼ How to Use find and cpio to Perform Backups ..................................................... 363
▼ How to Print a Copy of a Zone Configuration ....................................................... 364
Restoring a Non-Global Zone .................................................................................. 364
▼ How to Restore an Individual Non-Global Zone ................................................... 364

28 Upgrading a Solaris 10 System That Has Installed Non-Global Zones ............... 367
   Back up Your System Before Performing an Upgrade ........................................ 367
   Information on Upgrading a Solaris 10 System With Installed Zones .................. 367

29 Troubleshooting Miscellaneous Solaris Zones Problems .................................... 369
   Zones With an fs Resource Defined With a Type of lofs Cannot Be Upgraded to the Solaris 10 11/06 Release ................................................................. 369
   Solaris 10 6/06 and Solaris 10 11/06: Do Not Place the Root File System of a Non-Global Zone on ZFS ................................................................. 370
   Zone Administrator Mounting Over File Systems Populated by the Global Zone .... 370
   Zone Does Not Halt .............................................................................................. 371
   Incorrect Privilege Set Specified in Zone Configuration ....................................... 371
   netmasks Warning Displayed When Booting Zone .............................................. 371
   Resolving Problems With a zoneadm attach Operation ....................................... 372
      ▼ Patches and Packages Are Out of Sync ......................................................... 372
      ▼ Operating System Releases Do Not Match ................................................... 373

Glossary .................................................................................................................. 375

Index ...................................................................................................................... 379
Preface

System Administration Guide: Solaris Containers—Resource Management, and Solaris Zones 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. This document cites any implementation differences between the platform types.

About Solaris Containers

A Solaris Container is a complete runtime environment for applications. Solaris 10 Resource Manager and Solaris Zones software partitioning technology are both parts of the container. These components address different qualities the container can deliver and work together to create a complete 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.
Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems that run the Solaris 10 release. To use this book, you should have at least one to two years of UNIX® system administration experience.

How the System Administration Volumes Are Organized

Here is a list of the topics that are covered by the volumes of 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>Printing services, 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, IP filter, Mobile IP, IP network multipathing (IPMP), and IPQoS</td>
</tr>
<tr>
<td>System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)</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>System Administration Guide: Naming and Directory Services (NIS+)</td>
<td>NIS+ naming and directory services</td>
</tr>
<tr>
<td>System Administration Guide: Network Services</td>
<td>Web cache servers, time-related services, network file systems (NFS and Autofs), mail, SLP, and PPP</td>
</tr>
<tr>
<td>System Administration Guide: Security Services</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>System Administration Guide: Solaris Containers-Resource Management and Solaris Zones</td>
<td>Resource management topics projects and tasks, extended accounting, resource controls, fair share scheduler (FSS), physical memory control using the resource capping daemon (rcapd), and resource pools; virtualization using Solaris Zones software partitioning technology</td>
</tr>
</tbody>
</table>
**Documented, 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>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>
<tr>
<td>AaBbCc123</td>
<td>Book titles, new terms, and terms to be emphasized</td>
<td>Read Chapter 6 in the User's Guide. A cache is a copy that is stored locally. Do not save the file. <strong>Note:</strong> Some emphasized items appear bold online.</td>
</tr>
</tbody>
</table>
Shell Prompts in Command Examples

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

<table>
<thead>
<tr>
<th>Shell</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>C shell</td>
<td>machine_name%</td>
</tr>
<tr>
<td>C shell for superuser</td>
<td>machine_name#</td>
</tr>
<tr>
<td>Bourne shell and Korn shell</td>
<td>$</td>
</tr>
<tr>
<td>Bourne shell and Korn shell</td>
<td>#</td>
</tr>
</tbody>
</table>
This part introduces Solaris 10 Resource Manager, which enables you to control how applications use available system resources.
Introduction to Solaris 10 Resource Manager

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 25
- "When to Use Resource Management" on page 28
- "Setting Up Resource Management (Task Map)" on page 30

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 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 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 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 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. The extended accounting facility that links these identifiers with application resource usage is described in Chapter 4.

Interaction With Solaris Zones
Resource management features can be used with Solaris 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 <code>/etc/project</code> file, in the NIS map, or in the LDAP directory service.</td>
<td>“project Database” on page 36</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>“Monitoring by System” on page 192 and <code>cpustat(1M), iostat(1M), mpstat(1M), prstat(1M), sar(1), and vmstat(1M)</code> 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>“Available Resource Controls” on page 76, “Global and Local Actions on Resource Control Values” on page 81, “Temporarily Updating Resource Control Values on a Running System” on page 85 and <code>rctladm(1M)</code> and <code>prctl(1)</code> 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 <code>/etc/project</code> 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>“project Database” on page 36, “Local <code>/etc/project</code> File Format” on page 37, “Available Resource Controls” on page 76, “Global and Local Actions on Resource Control Values” on page 81, and Chapter 8</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 <code>rcap_max_rss</code> attribute in the <code>/etc/project</code> file.</td>
<td>“project Database” on page 36 and Chapter 10</td>
</tr>
</tbody>
</table>
### Setting Up Resource Management (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<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 36 and Chapter 12</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 112 and <code>dispadmin(1M)</code> man page</td>
</tr>
<tr>
<td>Activate the extended accounting facility to monitor and record</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 Processes, Tasks, and Flows&quot; on page 66 and <code>acctadm(1M)</code> man page</td>
</tr>
<tr>
<td>resource consumption on a task or process basis.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional) If you need to make additional adjustments to your</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 85 and <code>rctladm(1M)</code> and <code>prctl(1)</code> man pages</td>
</tr>
<tr>
<td>configuration, you can continue to alter the values from the</td>
<td></td>
<td></td>
</tr>
<tr>
<td>command line. You can alter the values while the task or process is running.</td>
<td></td>
<td></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 61</td>
</tr>
</tbody>
</table>
CHAPTER 2

Projects and Tasks (Overview)

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 34
- “Project Identifiers” on page 34
- “Task Identifiers” on page 40
- “Commands Used With Projects and Tasks” on page 41

To use the projects and tasks facilities, see Chapter 3.

What’s New in Project Database and Resource Control Commands for Solaris 10?

Enhancements include the following:

- Scaled value and unit modifier support for resource control values and commands
- Improved validation and easier manipulation of the project attributes field
- Revised output format and new options for the prctl and projects commands
- Ability to set user’s default project through the useradd command and modify information by using the usermod and passmgmt commands

In addition to the information contained in this chapter and Chapter 6, see the following man pages:

- passmgmt(1M)
- projadd(1M)
- projmod(1M)
- useradd(1M)
For a complete listing of new Solaris 10 features and a description of Solaris releases, see Solaris 10 What’s New.

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 settaskid 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 settaskid(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`
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 name 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 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.

The account management PAM module for projects is documented in the `pam_projects(5)` man page. For an overview of PAM, see Chapter 16, “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 (_) characters, hyphens (-), 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. projname cannot contain colons (:) or newline characters.

projid The project’s unique numerical ID (PROJID) within the system. The maximum value of the projid field is UID_MAX (2147483647).

comment A description of the project.

user-list 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.
group-list  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.

attributes  A semicolon-separated list of name-value pairs, such as resource controls (see
Chapter 6). name is an arbitrary string that specifies the object-related attribute,
and value is the optional value for that attribute.

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.

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 /etc/project file:

```
system:0:System::::
user.root:1:Super-User::::
noproject:2:No Project::::
default:3::::

```

group.staff:10::::

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 88.
- 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” on page 119.
- To add a `project.pool` attribute to a project’s entry, see “Creating the Configuration” on page 182.

### 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.

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>projects(1)</td>
<td>Displays project memberships for users. Lists projects from project database. Prints information on given projects. If no project names are supplied, information is displayed for all projects. Use the projects command with the -l option to print verbose output.</td>
</tr>
<tr>
<td>newtask(1)</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. newtask can also be used to change the task and the project binding for a running process. Use with the -F option to create a finalized task.</td>
</tr>
<tr>
<td>passmgmt(1M)</td>
<td>Updates information in the password files. Use with the -K key=value option to add to user attributes or replace user attributes in local files.</td>
</tr>
<tr>
<td>projadd(1M)</td>
<td>Adds a new project entry to the /etc/project file. The projadd command creates a project entry only on the local system. projadd cannot change information that is supplied by the network naming service. Can be used to edit project files other than the default file, /etc/project. Provides syntax checking for project file. Validates and edits project attributes. Supports scaled values.</td>
</tr>
<tr>
<td>projmod(1M)</td>
<td>Modifies information for a project on the local system. projmod 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, /etc/project. 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.</td>
</tr>
<tr>
<td>projdel(1M)</td>
<td>Deletes a project from the local system. projdel 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 -K key=value 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 -K key=value 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 44
- “Administering Projects” on page 47

For an overview of the projects and tasks facilities, see Chapter 2.

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 44</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 47</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 50</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:

```
# 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:

```
# 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
324 root 29M 75M sleep 59 0 0:08.27 0.2% Xsun/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
1845 jtd 24M 22M sleep 49 0 0:00.29 0.0% dtmail/11
1009 jtd 9864K 8384K sleep 49 0 0:00.59 0.0% dtwm/8
114 root 1640K 704K sleep 58 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 0 0 0:00.00 0.0% keyserv/4

PROJID NPROC SIZE RSS MEMORY TIME CPU PROJECT
10 52 200M 271M 68% 0:11.45 0.4% booksite
0 35 113M 129M 32% 0:10.46 0.2% system

Total: 87 processes, 205 \wps, 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
23023 root 26M 20M sleep 59 0 0:03:18 0.6% Xsun/1
```
Example Commands and Command Options

<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 lwp, 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 task id 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
```
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.**
   Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

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:::
     projid : 0
     comment: ""
     users : (none)
     groups : (none)
     attrs:
user.root
     projid : 1
     comment: ""
     users : (none)
     groups : (none)
     attrs:
noproject
     projid : 2
     comment: ""
     users : (none)
     groups : (none)
     attrs:
default
     projid : 3
     comment: ""
     users : (none)
     groups : (none)
     attrs:
group.staff
     projid : 10
     comment: ""
```
3 Add a project with the name *booksite*. Assign the project to a user who is named *mark* with project ID number 4113.

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

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

```
# 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.
   # projmod -c 'Book Auction Project' booksite

6  View the changes in 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:
   nobproject
     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 176.
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.
Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

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)
    attrs:
user.root
    projid : 1
    comment: ""
    users : (none)
    groups : (none)
    attrs:
noproject
    projid : 2
    comment: ""
    users : (none)
    groups : (none)
    attrs:
default
    projid : 3
    comment: ""
    users : (none)
    groups : (none)
    attrs:
group.staff
    projid : 10
    comment: ""
    users : (none)
    groups : (none)
    attrs:
```
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, as superuser, type the following:

# ypcat project | projmod -f –

Note – The ypcat project | projmod -f – command is not yet implemented.

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

# 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.

$ 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.

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.

   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.

   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.**
   Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

   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.**
   ```
   # pgrep book_catalog
   8100
   ```

3. **Associate process 8100 with a new task ID in the booksite project.**
   ```
   # 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.**
   ```
   # 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.
Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

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.

```
# projmod -a -K task.max-lwps myproject
```

3 You can then add a value to task.max-lwps in the project myproject. The value consists of a privilege level, a threshold value, and an action associated with reaching the threshold.

```
# 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.

```
# projmod -a -K "task.max-lwps=(priv,1000,signal=KILL)" myproject
```

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

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

How to Remove Attribute Values From Projects

This procedure assumes the values:

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

1 Become superuser or assume an equivalent role.
Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

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

```
# projmod -r -K "task.max-lwps=(priv,100,deny)" myproject
```
If `task.max-lwps` has multiple values, such as:

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

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

```
task.max-lwps=(priv,1000,signal=KILL)
```

▼ **How to Remove a Resource Control Attribute From a Project**

To remove the resource control `task.max-lwps` in the project `myproject`, use the `projmod` command with the `-r` and `-K` options.

1. **Become superuser or assume an equivalent role.**
   Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2. **Remove the attribute `task.max-lwps` and all of its values from the project `myproject`:**
   ```
   # projmod -r -K 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.**
   Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

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.
   Roles contain authorizations and privileged commands. For more information about roles, see "Using the Solaris Management Tools With RBAC (Task Map)" in System Administration Guide: Basic Administration.

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 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 58
- “How Extended Accounting Works” on page 58
- “Extended Accounting Configuration” on page 60
- “Commands Used With Extended Accounting” on page 60
- “Perl Interface to libexacct” on page 61

To begin using extended accounting, see “How to Activate Extended Accounting for Processes, Tasks, and Flows” on page 66.

What’s New in Extended Accounting for Solaris 10?

mstat e data for process accounting can now be generated. See “How to View Available Accounting Resources” on page 67.

For a complete listing of new Solaris 10 features and a description of Solaris releases, see Solaris 10 What’s New.
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 36, “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 on a task or process basis, or on the basis of selectors provided by the IPQoS flowacct module. For more information, see ipqos(7IPP).

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 exact 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 61.

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.
Extensible Format

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.

exactc Records and Format

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

- To enable third-party exactc 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 exact format. This library is the only supported interface to exact 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 330 for more information.

### Extended Accounting Configuration

The /etc/acctadm.conf file contains the current extended accounting configuration. The file is edited through the acctadm interface, not by the user.

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.

### 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, and flows.</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. lastcomm 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 44. 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 exacct framework. You can also create Perl scripts that write exacct 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 exacct-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:: prefix. All of the classes provided by the Perl exacct library are found under the Sun::Solaris::Exacct module.

The underlying libexacct(3LIB) library provides operations on exacct format files, catalog tags, and exacct objects. exacct 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 exacct module. This module provides functions to access the exacct-related system calls getacct(2), putacct(2), and wracct(2). This module also provides functions to access the libexacct(3LIB) library function ea_error(3EXACCT). Constants for all of the exacct EO_<em>, EW_</em>, EXR_<em>, P_</em>, and TASK_* macros are also provided in this module.</td>
<td>Exacct(3PERL)</td>
</tr>
<tr>
<td>Sun::Solaris::Exacct::Catalog</td>
<td>This module provides object-oriented methods to access the bitfields in an exacct catalog tag. This module also provides access to the constants for the EXC_<em>, EXD_</em>, and EXD_* macros.</td>
<td>Exacct::Catalog(3PERL)</td>
</tr>
<tr>
<td>Sun::Solaris::Exacct::File</td>
<td>This module provides object-oriented methods to access the libexacct accounting file functions ea_open(3EXACCT), ea_close(3EXACCT), ea_get_creator(3EXACCT), ea_get_hostname(3EXACCT), ea_next_object(3EXACCT), ea_previous_object(3EXACCT), and ea_write_object(3EXACCT).</td>
<td>Exacct::File(3PERL)</td>
</tr>
<tr>
<td>Sun::Solaris::Exacct::Object</td>
<td>This module provides object-oriented methods to access an individual exacct accounting file object. An exacct object is represented as an opaque reference blessed into the appropriate Sun::Solaris::Exacct::Object subclass. This module is further subdivided into the object types Item and Group. At this level, there are methods to access the ea_match_object_catalog(3EXACCT) and ea_attach_to_object(3EXACCT) functions.</td>
<td>Exacct::Object(3PERL)</td>
</tr>
<tr>
<td>Sun::Solaris::Exacct::Object::Item</td>
<td>This module provides object-oriented methods to access an individual exacct accounting file Item. Objects of this type inherit from Sun::Solaris::Exacct::Object.</td>
<td>Exacct::Object::Item(3PERL)</td>
</tr>
<tr>
<td>Sun::Solaris::Exacct::Object::Group</td>
<td>This module provides object-oriented methods to access an individual exacct accounting file Group. Objects of this type inherit from Sun::Solaris::Exacct::Object. These objects provide access to the ea_attach_to_group(3EXACCT) function. The Items contained within the Group are presented as a Perl array.</td>
<td>Exacct::Object::Group(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::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 libexacct" on page 68.
This chapter describes how to administer the extended accounting subsystem.

For an overview of the extending accounting subsystem, see Chapter 4.

## 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>“How to Activate Extended Accounting for Processes, Tasks, and Flows” on page 66, “How to Activate Extended Accounting With a Startup Script” on page 66</td>
</tr>
<tr>
<td>Display extended accounting status.</td>
<td>Determine the status of the extended accounting facility.</td>
<td>“How to Display Extended Accounting Status” on page 67</td>
</tr>
<tr>
<td>View available accounting resources.</td>
<td>View the accounting resources available on your system.</td>
<td>“How to View Available Accounting Resources” on page 67</td>
</tr>
<tr>
<td>Deactivate the process, task, and flow accounting facility.</td>
<td>Turn off the extended accounting functionality.</td>
<td>“How to Deactivate Process, Task, and Flow Accounting” on page 68</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>“Using the Perl Interface to libexacct” on page 68</td>
</tr>
</tbody>
</table>
How to Activate Extended Accounting for Processes, Tasks, and Flows

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

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Activate extended accounting for processes.

# 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

4 Activate extended accounting for flows.

# acctadm -e extended -f /var/adm/exacct/flow flow

See Also See acctadm(1M) for more information.

How to Activate Extended Accounting With a Startup Script

Activate extended accounting on an ongoing basis by linking the /etc/init.d/acctadm script into /etc/rc2.d.

# ln -s /etc/init.d/acctadm /etc/rc2.d/Snacctadm
# ln -s /etc/init.d/acctadm /etc/rc2.d/Knacctadm

The n variable is replaced by a number.

You must manually activate extended accounting at least once to set up the configuration.

See "Extended Accounting Configuration” on page 60 for information on accounting configuration.
How to Display Extended Accounting Status

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

```
# 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 much 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 resources available on your system.

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

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

Chapter 5 • Administering Extended Accounting (Tasks) 67
How to Deactivate Process, Task, and Flow Accounting

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

1. **Become superuser or assume an equivalent role.**
   Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

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. **Verify that task accounting, process accounting, and flow accounting have been turned off.**
   ```
   # acctadm
   Task accounting: inactive
   Task accounting file: none
   Tracked task resources: extended
   Untracked task resources: none
   Process accounting: inactive
   Process accounting file: none
   Tracked process resources: extended
   Untracked process resources: host
   Flow accounting: inactive
   Flow accounting file: none
   Tracked flow resources: extended
   Untracked flow resources: none
   ```

Using the Perl Interface to libexacct

How to Recursively Print the Contents of an exacct Object

Use the following code to recursively print the contents of an exacct 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.
sub dump_object
{
    my ($obj, $indent) = @_;  # Retrieve the catalog tag. Because we are
    my $istr = ' ' x $indent;  # 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 s 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("ITEM\nCatalog = %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 Exacct::Object::Group).
        # We could of course figure out that the item contained
        # a nested item orgroup 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);
        }
    }
}

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/perl5/5.6.1/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 => $$     ],
);
How to Print the Contents of an exacct File

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

```
#!/usr/perl5/5.6.1/bin/perl

cd("/path/to/exacct_file")

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

die("Usage is dumpexacct <exacct 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 ($obj = $ef->get()) {
    ea_dump_object($obj);
}

# Report any errors
if (ea_error() != EXR_OK && ea_error() != EXR_EOF) {
    # Error handling code here...
}
```
printf("\nERROR: \n", ea_error_str());
    exit(1);
}
exit(0);

Example Output From
Sun::Solaris::Exacct::Object-&gt;dump()

Here is example output produced by running Sun::Solaris::Exacct::Object-&gt;dump() on the file created in “How to Create a New Group Record and Write It to a File” on page 70.

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
ENDDGROUP
After you determine the resource consumption of workloads on your system as described in Chapter 4, 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 74
- “Configuring Resource Controls and Attributes” on page 76
- “Applying Resource Controls” on page 85
- “Temporarily Updating Resource Control Values on a Running System” on page 85
- “Commands Used With Resource Controls” on page 86

For information about how to administer resource controls, see Chapter 7.

What's New in Resource Controls for Solaris 10?

The following set of resource controls replaces the System V interprocess communication (IPC) /etc/system tunables:

- project.max-shm-ids
- project.max-msg-ids
- project.max-sem-ids
- project.max-shm-memory
- process.max-sem-nsems
- process.max-sem-ops
- process.max-msg-qbytes

The following event port resource controls have been added:

- project.max-device-locked-memory
- project.max-port-ids
The following cryptographic resource control has been added:

- `process.max-crypto-memory`

The following additional resource controls have been added:

- `project.max-lwps`
- `project.max-tasks`
- `project.max-contracts`

For more information, see "Available Resource Controls" on page 76.

For a complete listing of new Solaris 10 features and a description of Solaris releases, see *Solaris 10 What’s New*.

### 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. 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 `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.

On a system running zones, the prefix `zone` identifies a zone-wide resource control. See "Resource Type Properties" on page 223 for information on available zone-wide resource controls.

For a list of the standard resource controls that are available in this release, see "Available Resource Controls" on page 76.

### 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 96 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, and projects 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 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.

**Configuring Resource Controls and Attributes**

The resource controls facility is configured through the `project` database. See Chapter 2. 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 6–2.

```
development:101:Developers:::task.max-lwps=(privileged,10,deny);
    process.max-cpu-time=(basic,3600,signal=TERM),(priv,3660,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 219 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 81, `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 Resource Controls

<table>
<thead>
<tr>
<th>Control Name</th>
<th>Description</th>
<th>Default Unit</th>
</tr>
</thead>
<tbody>
<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 <code>FSS(7)</code>).</td>
<td>Quantity (shares)</td>
</tr>
<tr>
<td><code>project.max-crypto-memory</code></td>
<td>Total amount of kernel memory that can be used by <code>libpks11</code> 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><code>project.max-device-locked-memory</code></td>
<td>Total amount of locked memory allowed. This resource control might not be included in a future release.</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td><code>project.max-port-ids</code></td>
<td>Maximum allowable number of event ports.</td>
<td>Quantity (number of event ports)</td>
</tr>
<tr>
<td><code>project.max-shm-ids</code></td>
<td>Maximum number of shared memory IDs allowed for this project.</td>
<td>Quantity (shared memory IDs)</td>
</tr>
<tr>
<td><code>project.max-sem-ids</code></td>
<td>Maximum number of semaphore IDs allowed for this project.</td>
<td>Quantity (semaphore IDs)</td>
</tr>
<tr>
<td><code>project.max-msg-ids</code></td>
<td>Maximum number of message queue IDs allowed for this project.</td>
<td>Quantity (message queue IDs)</td>
</tr>
<tr>
<td><code>project.max-shm-memory</code></td>
<td>Total amount of shared memory allowed for this project.</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td><code>project.max-lwps</code></td>
<td>Maximum number of LWPs simultaneously available to this project.</td>
<td>Quantity (LWPs)</td>
</tr>
</tbody>
</table>
### TABLE 6–1  Standard Resource Controls  *(Continued)*

<table>
<thead>
<tr>
<th>Control Name</th>
<th>Description</th>
<th>Default Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>project.max-tasks</td>
<td>Maximum number of tasks allowable in this project.</td>
<td>Quantity (number of tasks)</td>
</tr>
<tr>
<td>project.max-contracts</td>
<td>Maximum number of contracts allowed in this project.</td>
<td>Quantity (contracts)</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 LWPs simultaneously available to this task’s processes.</td>
<td>Quantity (LWPs)</td>
</tr>
<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 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 available to this process.</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td>process.max-port-events</td>
<td>Maximum allowable number of events per event port.</td>
<td>Quantity (number of events)</td>
</tr>
<tr>
<td>process.max-sem-nsems</td>
<td>Maximum number of semaphores allowed per semaphore set.</td>
<td>Quantity (semaphores per set)</td>
</tr>
<tr>
<td>process.max-sem-ops</td>
<td>Maximum number of semaphore operations allowed per semop call (value copied from the resource control at semget() time).</td>
<td>Quantity (number of operations)</td>
</tr>
<tr>
<td>process.max-msg-qbytes</td>
<td>Maximum number of bytes of messages on a message queue (value copied from the resource control at msgget() time).</td>
<td>Size (bytes)</td>
</tr>
</tbody>
</table>
TABLE 6–1  Standard Resource Controls  (Continued)

<table>
<thead>
<tr>
<th>Control Name</th>
<th>Description</th>
<th>Default Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>process.max-msg-messages</td>
<td>Maximum number of messages on a message queue (value copied from the resource control at msgget() time).</td>
<td>Quantity (number of messages)</td>
</tr>
</tbody>
</table>

The following zone-wide resource controls are available on a Solaris system with zones installed:

- `zone.cpu-shares`
- `zone.max-lwps`

For information on zone-wide resource controls, see “Resource Type Properties” on page 223.

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.

### 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 `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>KB</td>
<td></td>
<td>$2^{10}$</td>
</tr>
<tr>
<td></td>
<td>MB</td>
<td></td>
<td>$2^{20}$</td>
</tr>
<tr>
<td></td>
<td>GB</td>
<td></td>
<td>$2^{30}$</td>
</tr>
<tr>
<td></td>
<td>TB</td>
<td></td>
<td>$2^{40}$</td>
</tr>
<tr>
<td></td>
<td>PB</td>
<td></td>
<td>$2^{50}$</td>
</tr>
<tr>
<td></td>
<td>EB</td>
<td></td>
<td>$2^{60}$</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.

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:
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.

signal=

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 6–2.

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 rctladm(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>
<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>
</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.

Use the `prctl` command to display local values and actions for the resource control.

```
$ 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</td>
<td>inf</td>
<td>signal=XCPU</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td></td>
<td>18.4Es</td>
<td>inf</td>
<td>none</td>
<td></td>
</tr>
</tbody>
</table>

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.

![Diagram of resource control enforcement](image)

**FIGURE 6-1** Process Collectives, Container Relationships, and Their Resource Control Sets

## 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 \texttt{syslog} action on a resource control. Then, if any entity managed by that resource control encounters a threshold value, a system message is logged at the configured logging level. See Chapter 7 and the \texttt{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 \texttt{newtask}, \texttt{su}, or the other project-aware launchers \texttt{at}, \texttt{batch}, or \texttt{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 \texttt{login(1)}, \texttt{newtask(1)}, \texttt{at(1)}, \texttt{cron(1M)}, and \texttt{su(1M)} for more information.

Updates to entries in the project database, whether to the \texttt{/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 \texttt{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 \texttt{rctladm} and \texttt{prctl} commands to update resource controls on a running system.

### Updating Logging Status

The \texttt{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 \texttt{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 \texttt{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><code>ipcs(1)</code></td>
<td>Allows you to observe which IPC objects are contributing to a project's usage</td>
</tr>
<tr>
<td><code>prctl(1)</code></td>
<td>Allows you to make runtime interrogations of and modifications to the resource controls facility, with local scope</td>
</tr>
<tr>
<td><code>rctladm(1M)</code></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.
Administering Resource Controls (Tasks)

This chapter describes how to administer the resource controls facility.

For an overview of the resource controls facility, see Chapter 6.

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 88</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 91</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 rctlad” on page 95</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 96</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.**
   Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

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)
     attrs:
     .
     .
     .
     x-files
     projid : 100
     comment: ""
     users : (none)
     ```
Example 7–1  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.

```
# 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.

Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Use the `projmod` command with the `-s` and `-K` options to set resource controls on project `x-files`:

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

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 88.
- 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:

- Type:

  ```
  # projects -l
  
  x-files
  
  projid: 100
  comment: ""
  users: (none)
  groups: (none)
  attribs: process.max-file-descriptor=(basic,128,deny); task.max-lwps=(basic,10,none),(privileged,500,deny)  one line in file
  
  Type:
  
  # cat etc/project
  
  x-files:100::::process.max-file-descriptor=(basic,128,deny);
  task.max-lwps=(basic,10,none),(privileged,500,deny)  one line in file
  ```
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></td>
<td></td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td></td>
<td>65.5K</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td></td>
<td>2.15G</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.crypto-buffer-limit</td>
<td></td>
<td></td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td></td>
<td>16.0EB</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-crypto-sessions</td>
<td></td>
<td></td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td></td>
<td>18.4E</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.add-crypto-sessions</td>
<td></td>
<td></td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td></td>
<td>100</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td></td>
<td>18.4E</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.min-crypto-sessions</td>
<td></td>
<td></td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td></td>
<td>20</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td></td>
<td>18.4E</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-msg-messages</td>
<td></td>
<td></td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td></td>
<td>8.19K</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td></td>
<td>4.29G</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-msg-qbytes</td>
<td></td>
<td></td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td></td>
<td>64.0KB</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td></td>
<td>16.0EB</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-sem-ops</td>
<td></td>
<td></td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td></td>
<td>512</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td></td>
<td>2.15G</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-sem-nsems</td>
<td></td>
<td></td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td></td>
<td>512</td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td></td>
<td>32.8K</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>process.max-address-space</td>
<td></td>
<td></td>
<td>-</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>privileged</td>
<td></td>
<td>16.0EB</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
<tr>
<td>system</td>
<td></td>
<td>16.0EB</td>
<td>max</td>
<td>deny</td>
<td>-</td>
</tr>
</tbody>
</table>
### Using the prctl Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>process.max-file-descriptor</td>
<td></td>
</tr>
<tr>
<td>basic</td>
<td>256</td>
</tr>
<tr>
<td>privileged</td>
<td>65.5K</td>
</tr>
<tr>
<td>system</td>
<td>2.15G</td>
</tr>
<tr>
<td>process.max-core-size</td>
<td></td>
</tr>
<tr>
<td>privileged</td>
<td>8.00EB</td>
</tr>
<tr>
<td>system</td>
<td>8.00EB</td>
</tr>
<tr>
<td>process.max-stack-size</td>
<td></td>
</tr>
<tr>
<td>basic</td>
<td>8.00MB</td>
</tr>
<tr>
<td>privileged</td>
<td>8.00EB</td>
</tr>
<tr>
<td>system</td>
<td>8.00EB</td>
</tr>
<tr>
<td>process.max-data-size</td>
<td></td>
</tr>
<tr>
<td>privileged</td>
<td>16.0EB</td>
</tr>
<tr>
<td>system</td>
<td>16.0EB</td>
</tr>
<tr>
<td>process.max-file-size</td>
<td></td>
</tr>
<tr>
<td>privileged</td>
<td>8.00EB, signal=XFSZ</td>
</tr>
<tr>
<td>system</td>
<td>8.00EB</td>
</tr>
<tr>
<td>process.max-cpu-time</td>
<td></td>
</tr>
<tr>
<td>privileged</td>
<td>18.4Es, signal=XCPU</td>
</tr>
<tr>
<td>system</td>
<td>18.4Es, none</td>
</tr>
<tr>
<td>task.max-cpu-time</td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>18.4Es, none</td>
</tr>
<tr>
<td>task.max-lwps</td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>2.15G</td>
</tr>
<tr>
<td>project.max-contracts</td>
<td></td>
</tr>
<tr>
<td>privileged</td>
<td>10.0K</td>
</tr>
<tr>
<td>system</td>
<td>2.15G</td>
</tr>
<tr>
<td>project.max-device-locked-memory</td>
<td></td>
</tr>
<tr>
<td>privileged</td>
<td>499MB</td>
</tr>
<tr>
<td>system</td>
<td>16.0EB</td>
</tr>
<tr>
<td>project.max-port-ids</td>
<td></td>
</tr>
<tr>
<td>privileged</td>
<td>8.19K</td>
</tr>
<tr>
<td>system</td>
<td>65.5K</td>
</tr>
<tr>
<td>project.max-shm-memory</td>
<td></td>
</tr>
<tr>
<td>privileged</td>
<td>1.95GB</td>
</tr>
<tr>
<td>system</td>
<td>16.0EB</td>
</tr>
<tr>
<td>project.max-shm-ids</td>
<td></td>
</tr>
<tr>
<td>privileged</td>
<td>128</td>
</tr>
<tr>
<td>system</td>
<td>16.8M</td>
</tr>
<tr>
<td>project.max-msg-ids</td>
<td></td>
</tr>
<tr>
<td>privileged</td>
<td>128</td>
</tr>
<tr>
<td>system</td>
<td>16.8M</td>
</tr>
<tr>
<td>project.max-sem-ids</td>
<td></td>
</tr>
<tr>
<td>privileged</td>
<td>128</td>
</tr>
<tr>
<td>system</td>
<td>16.8M</td>
</tr>
<tr>
<td>project.max-tasks</td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>2.15G</td>
</tr>
<tr>
<td>project.max-lwps</td>
<td></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>deny</td>
<td>-</td>
<td>110453</td>
</tr>
<tr>
<td></td>
<td>privileged</td>
<td>65.5K</td>
<td>deny</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>system</td>
<td>2.15G</td>
<td>deny</td>
<td>-</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 88.

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

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
   process: 111108: csh
   NAME PRIVILEGE VALUE FLAG ACTION RECIPIENT
   project.max-lwps privileged 3 - deny -
   system 2.15G max deny -

▼ **How to Use`prctl` to Lower a Resource Control Value**

1 Become superuser or assume an equivalent role.
   Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2 Use the `prctl` command with the `-r` option to change the lowest value of the `process.max-file-descriptor` resource control.
   # 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.
   Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in *System Administration Guide: Basic Administration*.

2 Display the value of `project.cpu-shares` in the project `group.staff`.
   # 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.
   # 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`.

```
# prctl -n project.cpu-shares -i project group.staff
```

<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>project.cpu-shares</td>
<td>privileged</td>
<td>10</td>
<td>none</td>
<td></td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>system</td>
<td>65.5K</td>
<td>max</td>
<td>none</td>
<td></td>
</tr>
</tbody>
</table>

**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 ]
```

Chapter 7 • Administering Resource Controls (Tasks) 95
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:
 m 3600 0 --rw--rw-- name staff x-files
 m 201 0 --rw--rw-- name staff x-files
 m 1802 0 --rw--rw-- name staff x-files
 m 503 0 --rw--rw-- name staff x-files
 m 304 0 --rw--rw-- name staff x-files
 m 605 0 --rw--rw-- name staff x-files
 m 6 0 --rw--rw-- name staff x-files
 m 107 0 --rw--rw-- name staff x-files
Semaphores:
 s 0 0 --rw--rw-- name 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.

   ```bash
   # 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.

   ```bash
   # 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 kern.notice]
   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 99
- “CPU Share Definition” on page 100
- “CPU Shares and Process State” on page 101
- “CPU Share Versus Utilization” on page 101
- “CPU Share Examples” on page 101
- “FSS Configuration” on page 103
- “FSS and Processor Sets” on page 105
- “Combining FSS With Other Scheduling Classes” on page 107
- “Setting the Scheduling Class for the System” on page 108
- “Scheduling Class on a System with Zones Installed” on page 108
- “Commands Used With FSS” on page 109

To begin using the fair share scheduler, see Chapter 9.

**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 give every process relatively equal access to the available CPUs. However, 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 system with zones installed, see “Zone Configuration Data” on page 219, “Resource Controls Used in Non-Global Zones” on page 328, and “Using the Fair Share Scheduler on a Solaris System With Zones Installed” on page 359.

## 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.

The following formula shows how the FSS scheduler calculates per-project allocation of CPU resources.

$$\text{allocation}_{\text{project}i} = \frac{\text{shares}_{\text{project}i}}{\sum_{j=1}^{n} (\text{shares}_{\text{project}j})}$$

$j$ is the index among all active projects

FIGURE 8–1 FSS Scheduler Share Calculation

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.

<table>
<thead>
<tr>
<th>Project A (1 share)</th>
<th>Project B (3 shares)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25%</td>
<td>75%</td>
</tr>
</tbody>
</table>

**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` database and name services, see "project Database" on page 36.

**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 the 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, as shown in the following figure.

\[
\text{allocation}_{\text{project } i} = \frac{\text{shares}_{\text{project } i}}{\text{processor set } X} \sum_{j=1}^{n} (\text{shares}_{\text{project } j})
\]

\(j\) is the index among all active projects that run on processor set \(X\)

**FIGURE 8–2** FSS Scheduler Share Calculation With Processor Sets
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.

<table>
<thead>
<tr>
<th>Processor Set #1</th>
<th>Processor Set #2</th>
<th>Processor Set #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 CPUs</td>
<td>4 CPUs</td>
<td>2 CPUs</td>
</tr>
<tr>
<td>25% of the system</td>
<td>50% of the system</td>
<td>25% of the system</td>
</tr>
<tr>
<td>Project A 16.66% (1/6)</td>
<td>Project B 40% (2/5)</td>
<td>Project C 100% (3/3)</td>
</tr>
<tr>
<td>Project B 33.33% (2/6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project C 50% (3/6)</td>
<td>Project C 60% (3/5)</td>
<td></td>
</tr>
</tbody>
</table>

The total system-wide project CPU allocations on such a system are shown in the following table.
Combining FSS With Other Scheduling Classes

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>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 * 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>

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...
Setting the Scheduling Class for the System

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 113 and `dispadmin(1M)`. To move running processes into a different scheduling class, see "Configuring the FSS" on page 112 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 booted or rebooted.

It is also possible to set the scheduling class for a non-global zone through the dynamic resource pools facility. An association made through a resource pool will override the system setting. See Step 6 in "How to Configure the Zone" on page 234 for information about setting the default scheduling class for a non-global zone by using the pools facility.

For information about moving running processes into a different scheduling class without changing the default scheduling class and rebooting, see Table 26–4 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>
Administering the Fair Share Scheduler (Tasks)

This chapter describes how to use the fair share scheduler (FSS).

For an overview of the FSS, see Chapter 8.

**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>“Monitoring the FSS” on page 112</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>“How to Make FSS the Default Scheduler Class” on page 113</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>“How to Manually Move Processes From the TS Class Into the FSS Class” on page 113</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>“How to Manually Move Processes From All User Classes Into the FSS Class” on page 114</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>“How to Manually Move a Project’s Processes Into the FSS Class” on page 114</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 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.
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.
Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

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
   ```

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.
Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

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.
   Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

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

3 Move all processes from their current scheduling classes into the FSS scheduling class.
   # 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.
   Roles contain authorizations and privileged commands. For more information about roles, see “Using the Solaris Management Tools With RBAC (Task Map)” in System Administration Guide: Basic Administration.

2 Move processes that run in project ID 10 to the FSS scheduling class.
   # 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
# RES=1000
# # Time Quantum
# QUANTUM=110
```

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.

The following topics are covered in this chapter.

- “Introduction to the Resource Capping Daemon” on page 118
- “How Resource Capping Works” on page 118
- “Attribute to Limit Physical Memory Usage” on page 119
- “rcapd Configuration” on page 119
- “Monitoring Resource Utilization With rcapstat” on page 123
- “Commands Used With rcapd” on page 124

For procedures using the rcapd feature, see Chapter 11.

What’s New in Physical Memory Control Using the Resource Capping Daemon?

**Solaris 10:** You can now use the projmod command to set the rcap.max-rss attribute in the /etc/project file.

**Solaris 10 11/06:** Information on enabling and disabling the resource capping daemon as a service in the Solaris service management facility (SMF) has been added.

For a complete listing of new Solaris 10 features and a description of Solaris releases, see *Solaris 10 What’s New.*
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 and the project(4) man page. For information about resource controls, see Chapter 6.

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 123 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

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:

\[ \text{rcap.max-rss} \] 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 gigabytes 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 use the \texttt{projmod} command to set the \texttt{rcap.max-rss} attribute in the /etc/project file:

```
# projmod -s -K rcap.max-rss=10GB db
```

The /etc/project file then contains the line:

```
db:100::db,root::rcap.max-rss=10737418240
```

**rcapd Configuration**

You use the \texttt{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 \texttt{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 Process Management role and the System Administrator role both include the Process Management profile.

Configuration changes can be incorporated into \texttt{rcapd} according to the configuration interval (see "\texttt{rcapd Operation Intervals} on page 122") or on demand by sending a \texttt{SIGHUP} (see the \texttt{kill(1)} man page).

If used without arguments, \texttt{rcapadm} displays the current status of the resource capping daemon if it has been configured.
The following subsections discuss cap enforcement, cap values, and \texttt{rcapd} operation intervals.

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

Solaris resource capping is not specifically zone-aware. Running \texttt{rcapd} in the global zone only enforces memory caps on global zone processes.

If you are using \texttt{rcapd} on a system with zones installed, you must add a project entry and configure the daemon in each zone where you want the daemon to run. \texttt{rcapd} will not act on processes in zones other than the one in which it is running.

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.

**Memory Cap Enforcement Threshold**

The \textit{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 128.
**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 `vm` column of `rcapstat` output. See "Monitoring Resource Utilization With *rcapstat*" on page 123 and the `rcapstat(1)` man page for more information.

**Reporting on shared memory**

The daemon cannot determine which pages of memory are shared with other processes or which are mapped multiple times within the same process. Since *rcapd* assumes that each page is unique, this results in a discrepancy between the reported (estimated) RSS and the actual RSS.
Certain workloads, such as databases, use shared memory extensively. For these workloads, you can sample a project’s regular usage to determine a suitable initial cap value. Use output from the `prstat` command with the `-J` option. See the `prstat(1M)` man page.

**rcapd Operation Intervals**

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 <code>rcapstat</code> is not current.</td>
</tr>
<tr>
<td>config</td>
<td>60</td>
<td>Number of seconds between reconfigurations. In a reconfiguration event, <code>rcapadm</code> reads the configuration file for updates, and scans the project database for new or revised project caps. Sending a <code>SIGHUP</code> to <code>rcapd</code> causes an immediate reconfiguration.</td>
</tr>
</tbody>
</table>

To tune intervals, see “How to Set Operation Intervals” on page 129.
Determining \texttt{rcapd} Scan Intervals

The scan interval controls how often \texttt{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, \texttt{rcapd} might not attribute the processes to a workload if the scan interval is lengthened.

Determining Sample Intervals

The sample interval configured with \texttt{rcapadm} is the shortest amount of time \texttt{rcapd} waits between sampling a workload’s usage and enforcing the cap if it is exceeded. If you reduce this interval, \texttt{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 \texttt{rcapstat} is shorter than the interval specified to \texttt{rcapd} with \texttt{rcapadm}, the output for some intervals can be zero. This situation occurs because \texttt{rcapd} does not update statistics more frequently than the interval specified with \texttt{rcapadm}. The interval specified with \texttt{rcapadm} is independent of the sampling interval used by \texttt{rcapstat}.

Monitoring Resource Utilization With \texttt{rcapstat}

Use \texttt{rcapstat} to monitor the resource utilization of capped projects. To view an example \texttt{rcapstat} report, see "Producing Reports With \texttt{rcapstat}" on page 130.

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

\begin{itemize}
  \item \textit{interval} \hspace{2cm} Specifies the sampling interval in seconds. The default interval is 5 seconds.
  \item \textit{count} \hspace{2cm} Specifies the number of times that the statistics are repeated. By default, \texttt{rcapstat} reports statistics until a termination signal is received or until the \texttt{rcapd} process exits.
\end{itemize}

The paging statistics in the first report issued by \texttt{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 \texttt{rcapstat} report.
<table>
<thead>
<tr>
<th>rcapstat 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>
<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&quot; on page 119 or the rcapd(1M) man 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 122.</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 122.</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>Command Reference</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</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>
<tr>
<td>rcapd(1M)</td>
<td>The resource capping daemon.</td>
</tr>
</tbody>
</table>
This chapter contains procedures for configuring and using the resource capping daemon rcapd.

For an overview of rcapd, see Chapter 10.

### 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>“How to Set the Memory Cap Enforcement Threshold” on page 128</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>“How to Set Operation Intervals” on page 129</td>
</tr>
<tr>
<td>Enable resource capping.</td>
<td>Activate resource capping on your system.</td>
<td>“How to Enable Resource Capping” on page 129</td>
</tr>
<tr>
<td>Disable resource capping.</td>
<td>Deactivate resource capping on your system.</td>
<td>“How to Disable Resource Capping” on page 130</td>
</tr>
<tr>
<td>Report cap and project information.</td>
<td>View example commands for producing reports.</td>
<td>“Reporting Cap and Project Information” on page 130</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>“Monitoring the RSS of a Project” on page 131</td>
</tr>
</tbody>
</table>
## Administering the Resource Capping Daemon With `rcapadm`

This section contains procedures for configuring the resource capping daemon with `rcapadm`. See “`rcapd` Configuration” on page 119 and the `rcapadm(1M)` man page for more information.

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

### How to Set the Memory Cap Enforcement Threshold

Caps can be configured so that they will not be enforced until the physical memory available to processes is low. See “Memory Cap Enforcement Threshold” on page 120 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.

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 `-c` option of `rcapadm` to set a different physical memory utilization value for memory cap enforcement.**
   
   ```
   # rcapadm -c percent
   ```

   `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.

**See Also**

To display the current physical memory utilization and the cap enforcement threshold, see “Reporting Memory Utilization and the Memory Cap Enforcement Threshold” on page 133.
How to Set Operation Intervals

“rcapd Operation Intervals” on page 122 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, type:

   ```
   # 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 the `svcadm` command or the `rcapadm` command with the `-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.

Producing Reports With `rcapstat`

Use `rcapstat` to report resource capping statistics. "Monitoring Resource Utilization With `rcapstat`” on page 123 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>50M</td>
<td>0K</td>
<td>3312K</td>
<td>0K</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>0K</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 shows project user1, which has an RSS in excess of its RSS cap.

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>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6144M</td>
<td>6144M</td>
<td>690M</td>
<td>220M</td>
<td>5528K</td>
<td>2764K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6144M</td>
<td>6144M</td>
<td>0M</td>
<td>131M</td>
<td>4912K</td>
<td>1637K</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 rcapd is consistently paging out memory as it attempts to meet the cap by lowering the physical memory utilization of the project's processes. However, rcapd 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, \texttt{rcapd} 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 \texttt{rcapd} to sample the workload and enforce caps more frequently.

\textbf{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 \texttt{user1} project is using more physical memory than its cap allows. This example shows how much memory the project workload requires.

```bash
user1machine% \texttt{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>6144M</td>
<td>690M</td>
<td>0K</td>
<td>689M</td>
<td>0K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6144M</td>
<td>6144M</td>
<td>0K</td>
<td>0K</td>
<td>0K</td>
<td>0K</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>0K</td>
<td>27M</td>
<td>0K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6146M</td>
<td>6144M</td>
<td>4872K</td>
<td>0K</td>
<td>4816K</td>
<td>0K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6150M</td>
<td>6144M</td>
<td>12M</td>
<td>0K</td>
<td>12M</td>
<td>0K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6150M</td>
<td>6144M</td>
<td>27M</td>
<td>0K</td>
<td>27M</td>
<td>0K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6150M</td>
<td>6144M</td>
<td>27M</td>
<td>0K</td>
<td>27M</td>
<td>0K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6150M</td>
<td>6144M</td>
<td>27M</td>
<td>0K</td>
<td>27M</td>
<td>0K</td>
</tr>
<tr>
<td>376565</td>
<td>user1</td>
<td>3</td>
<td>6249M</td>
<td>6150M</td>
<td>6144M</td>
<td>27M</td>
<td>0K</td>
<td>27M</td>
<td>0K</td>
</tr>
</tbody>
</table>

Halfway through the cycle, the cap on the \texttt{user1} project was increased from 6 gigabytes to 10 gigabytes. 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 \texttt{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 user 1 is 6 gigabytes, in every 5–second sample interval the RSS decreases and I/O increases as rcapd 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 gigabytes, approximately halfway through the example. The RSS then stabilizes at 6.1 gigabytes. 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 gigabytes 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 rcapstat 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.

```
376565  rcap  0  0K  0K  10G  0K  0K  0K
physical memory utilization: 55%  cap enforcement threshold: 0%
```

```
376565  rcap  0  0K  0K  10G  0K  0K  0K
physical memory utilization: 55%  cap enforcement threshold: 0%
```
Resource Pools (Overview)

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

Starting with the Solaris 10 11/06 release, resource pools and dynamic resource pools are now 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 136
- “Introduction to Dynamic Resource Pools” on page 137
- “About Enabling and Disabling Resource Pools and Dynamic Resource Pools” on page 137
- “Resource Pools Used in Zones” on page 137
- “When to Use Pools” on page 138
- “Resource Pools Framework” on page 139
- “Implementing Pools on a System” on page 141
- “project.pool Attribute” on page 141
- “SPARC: Dynamic Reconfiguration Operations and Resource Pools” on page 142
- “Creating Pools Configurations” on page 142
- “Directly Manipulating the Dynamic Configuration” on page 143
- “poold Overview” on page 143
- “Managing Dynamic Resource Pools” on page 144
- “Configuration Constraints and Objectives” on page 144
- “poold Features That Can Be Configured” on page 149
- “How Dynamic Resource Allocation Works” on page 151
- “Using poolstat to Monitor the Pools Facility and Resource Utilization” on page 154
- “Commands Used With the Resource Pools Facility” on page 156

For procedures using this functionality, see Chapter 13.
What's New in Resource Pools and Dynamic Resource Pools?

**Solaris 10**: Resource pools now provide a mechanism for adjusting each pool’s resource allocation in response to system events and application load changes. Dynamic resource pools 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.

You can now use the `projmod` command to set the `project.pool` attribute in the `/etc/project` file.

For a complete listing of new Solaris 10 features and a description of Solaris releases, see *Solaris 10 What's New*.

**Solaris 10 11/06**: Resource pools and dynamic resource pools are now SMF services.

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.

![Resource Pool Framework](image)

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:

```plaintext
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 `/etc/pool` 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 159.

**Resource Pools Used in Zones**

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 233.
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 poolsstat 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.

For information about resource pool commands, see “Commands Used With the Resource Pools Facility” on page 156.

When to Use Pools

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

Batch compute server
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.

Application or database server
Partition the resources for interactive applications in accordance with the applications’ requirements.

Turning on applications in phases
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.

Complex timesharing server
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.

Workloads that change seasonally

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 multiple pools configurations by invoking pooladm from a cron job. (See “Resource Pools Framework” on page 139.)

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 142.

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 database.

```plaintext
//etc/pooladm.conf
```

---

About Enabling and Disabling Resource Pools and Dynamic Resource Pools
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 159. 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 poold 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 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
```
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 poolsd control while the daemon is active. For more information, see “Identifying a Resource Shortage” on page 152.

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 can 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 the previous release.

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 174.

Note that the resource transfer might trigger action from `poold`. See "poold Overview" on page 143 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, `poolsd` can be stopped with the `SIGQUIT` or `SIGTERM` signal. Either of these signals causes `poolsd` to terminate gracefully.
- Although `poolsd` 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, `poolsd` acts on directions that you provide. You specify these directions as a series of constraints and objectives. `poolsd` uses your specifications to determine the relative value of different configuration possibilities in relation to the existing configuration. `poolsd` 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

See the `libpool(3LIB)` man page and "Pools Properties" on page 140 for more information about pools properties.

`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>wt-load</td>
<td>system</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>locality</td>
<td>pset</td>
<td>N/A</td>
<td>loose</td>
</tr>
</tbody>
</table>
Objectives are stored in property strings in the \libpool configuration. The property names are as follows:
- `system.poold.objectives`
- `pset.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 170 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.
If set, configurations that minimize resource locality are favored.

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 a > 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, `pooled` 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  pooled Objectives Example**

```
pset.pooled.objectives "utilization > 30; utilization < 80; locality tight"
```

See “How to Define Configuration Objectives” on page 170 for additional usage examples.
**poold Properties**

There are four categories of properties:
- Configuration
- Constraint
- Objective
- Objective Parameter

<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.

**poold Monitoring Interval**

Use the property name `system.poold.monitor-interval` to specify a value in milliseconds.

**poold 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.poold.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 poold 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 poold 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 151 for more information about using syslog.

For more information about how to configure poold logging, see "How to Set the poold Logging Level" on page 172.
Configuration Information Logging

The following types of messages can be generated:

- **ALERT** Problems accessing the `libpool` configuration, or some other fundamental, unanticipated failure of the `libpool` facility. Causes the daemon to exit and requires immediate administrative attention.

- **CRIT** Problems due to unanticipated failures. Causes the daemon to exit and requires immediate administrative attention.

- **ERR** Problems with the user-specified parameters that control operation, such as unresolvable, conflicting utilization objectives for a resource set. Requires administrative intervention to correct the objectives. `poold` attempts to take remedial action by ignoring conflicting objectives, but some errors will cause the daemon to exit.

- **WARNING** Warnings related to the setting of configuration parameters that, while technically correct, might not be suitable for the given execution environment. An example is marking all CPU resources as pinned, which means that `poold` cannot move CPU resources between processor sets.

- **DEBUG** Messages containing the detailed information that is needed when debugging configuration processing. This information is not generally used by administrators.

Monitoring Information Logging

The following types of messages can be generated:

- **CRIT** Problems due to unanticipated monitoring failures. Causes the daemon to exit and requires immediate administrative attention.

- **ERR** Problems due to unanticipated monitoring error. Could require administrative intervention to correct.

- **NOTICE** Messages about resource control region transitions.

- **INFO** Messages about resource utilization statistics.

- **DEBUG** Messages containing the detailed information that is needed when debugging monitoring processing. This information is not generally used by administrators.

Optimization Information Logging

The following types of messages can be generated:

- **WARNING** 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.

NOTICE Messages about usable configurations or configurations that will not be implemented due to overriding decision histories could be displayed.

INFO Messages about alternate configurations considered could be displayed.

DEBUG Messages containing the detailed information that is needed when debugging optimization processing. This information is not generally used by administrators.

**Logging Location**

The system. pool.l.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 144.

## 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**

poold 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 144 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 poold 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 149.

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 178.

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:

<table>
<thead>
<tr>
<th>id</th>
<th>Pool ID.</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool</td>
<td>Pool name.</td>
</tr>
<tr>
<td>rid</td>
<td>Resource set ID.</td>
</tr>
<tr>
<td>rset</td>
<td>Resource set name.</td>
</tr>
<tr>
<td>type</td>
<td>Resource set type.</td>
</tr>
<tr>
<td>min</td>
<td>Minimum resource set size.</td>
</tr>
<tr>
<td>max</td>
<td>Maximum resource set size.</td>
</tr>
<tr>
<td>size</td>
<td>Current resource set size.</td>
</tr>
<tr>
<td>used</td>
<td>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 (-).</td>
</tr>
<tr>
<td>load</td>
<td>Absolute representation of the load that is put on the resource set. For more information about this property, see the \libpool(3LIB) man page.</td>
</tr>
</tbody>
</table>

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.
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 137.

<table>
<thead>
<tr>
<th>Man Page Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pooladm(1M)</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>poolbind(1M)</td>
<td>Enables the manual binding of projects, tasks, and processes to a resource pool.</td>
</tr>
<tr>
<td>poolcfg(1M)</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>poold(1M)</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>poolstat(1M)</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.

**Administering Dynamic 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 159</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 159</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 139.</td>
<td>“How to Create a Static Configuration” on page 163</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 165</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 167</td>
</tr>
</tbody>
</table>
### Administering Dynamic Resource Pools (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set configuration constraints and define configuration objectives.</strong></td>
<td>Specify objectives for <code>pool</code> to consider when taking corrective action. For more information on configuration objectives, see “pool Overview” on page 143.</td>
<td>“How to Set Configuration Constraints” on page 169 and “How to Define Configuration Objectives” on page 170</td>
</tr>
<tr>
<td><strong>Set the logging level.</strong></td>
<td>Specify the level of logging information that <code>pool</code> generates.</td>
<td>“How to Set the <code>pool</code> Logging Level” on page 172</td>
</tr>
<tr>
<td><strong>Use a text file with the <code>poolcfg</code> command.</strong></td>
<td>The <code>poolcfg</code> command can take input from a text file.</td>
<td>“How to Use Command Files With <code>poolcfg</code>” on page 173</td>
</tr>
<tr>
<td><strong>Transfer resources in the kernel.</strong></td>
<td>Transfer resources in the kernel.</td>
<td>“Transferring Resources” on page 174</td>
</tr>
<tr>
<td><strong>Activate a pools configuration.</strong></td>
<td>Activate the configuration in the default configuration file.</td>
<td>“How to Activate a Pools Configuration” on page 175</td>
</tr>
<tr>
<td><strong>Validate a pools configuration before you commit the configuration.</strong></td>
<td>Validate a pools configuration to test what will happen when the validation occurs.</td>
<td>“How to Validate a Configuration Before Committing the Configuration” on page 175</td>
</tr>
<tr>
<td><strong>Remove a pools configuration from your system.</strong></td>
<td>All associated resources, such as processor sets, are returned to their default status.</td>
<td>“How to Remove a Pools Configuration” on page 175</td>
</tr>
<tr>
<td><strong>Bind processes to a pool.</strong></td>
<td>Manually associate a running process on your system with a resource pool.</td>
<td>“How to Bind Processes to a Pool” on page 176</td>
</tr>
<tr>
<td><strong>Bind tasks or projects to a pool.</strong></td>
<td>Associate tasks or projects with a resource pool.</td>
<td>“How to Bind Tasks or Projects to a Pool” on page 177</td>
</tr>
<tr>
<td><strong>Bind new processes to a resource pool.</strong></td>
<td>To automatically bind new processes in a project to a given pool, add an attribute to each entry in the project database.</td>
<td>“How to Set the <code>project.pool</code> Attribute for a Project” on page 177</td>
</tr>
<tr>
<td><strong>Use project attributes to bind a process to a different pool.</strong></td>
<td>Modify the pool binding for new processes that are started.</td>
<td>“How to Use project Attributes to Bind a Process to a Different Pool” on page 177</td>
</tr>
<tr>
<td><strong>Use the <code>poolstat</code> utility to produce reports.</strong></td>
<td>Produce multiple reports at specified intervals.</td>
<td>“Producing Multiple Reports at Specific Intervals” on page 179</td>
</tr>
</tbody>
</table>
Enabling and Disabling the Pools Facility

Starting with the Solaris 11/06 release, 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.

▼ Solaris 11/06: 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 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 pools service.
   # svcadm enable system/pools:default

▼ Solaris 11/06: 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 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 pools service.
   # svcadm disable system/pools:default
Solaris 10 11/06: How to Enable the Dynamic Resource Pools Service Using `svcadm`

1. Become superuser, or assume a role that includes the Service Management rights profile.
   Roles contain authorizations and privileged commands. For information on how to create the role and assign the role to a user, see “Configuring RBAC (Task Map)” in `System Administration Guide: Security Services Managing RBAC (Task Map)` in `System Administration Guide: Security Services`.

2. Enable the dynamic resource pools service.
   ```bash
   # svcadm enable system/pools/dynamic:default
   ```


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:

```bash
# 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:

```bash
# 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:

```bash
# 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:

```
# svcs -x *pool*
svc:/system/pools:default (resource pools framework)
  State: disabled since Wed 25 Jan 2006 10:41:05 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 10:41:12 AM GMT
  Reason: Service svc:/system/pools:default is disabled.
  See: http://sun.com/msg/SMF-8000-GE
  See: pooladm(1M)
  See: /var/svc/log/system-pools-dynamic:default.log
Impact: This service is not running.
```

Resource pools must be started for DRP to work. For example, resource pools could be started by using the pooladm command with the -e option:

```
# pooladm -e
```

Then the svcs *pool* command displays:

```
# svcs *pool*
STATE STIME FMRI
online 10:42:23 svc:/system/pools:default
online 10:42:24 svc:/system/pools/dynamic:default
```

▼ Solaris 11 06: 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 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 dynamic resource pools service.
   # 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 information on
how to create the role and assign the role to a user, see Managing RBAC (Task Map) in System

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 information on
how to create the role and assign the role to a user, see Managing 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 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. **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
   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
   ```
Commit the configuration at /etc/pooladm.conf.
# pooladm -c

(Optional) To copy the dynamic configuration to a static configuration file called /tmp/backup, type the following:
# pooladm -s /tmp/backup

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 information on how to create the role and assign the role to a user, see “Managing RBAC” in System Administration Guide: Security Services.

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.pooled.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

  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
```
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).

6 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  **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
    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
```
Commit the configuration at /etc/pooladm.conf:

```
# pooladm -c
```

(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 information on
how to create the role and assign the role to a user, see “Managing RBAC” in System

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 information on
how to create the role and assign the role to a user, see “Managing RBAC” in System

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")'  
   ```

4 Disable the locality objective for the pset_batch processor set.
   
   ```
   # poolcfg -c 'modify pset pset_batch (string pset.poold.objectives="locality none")'  
   ```

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
Commit the configuration at /etc/pooladm.conf.

# pooladm -c

(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 149 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 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 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 149. 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 information on how to create the role and assign the role to a user, see “Managing RBAC” in *System Administration Guide: Security Services*.

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 information on how to create the role and assign the role to a user, see “Managing RBAC” in System Administration Guide: Security Services.

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'
   ```

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)"
```

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 information on how to create the role and assign the role to a user, see “Managing RBAC” in *System Administration Guide: Security Services*.

2. Commit the configuration at `/etc/pooladm.conf`.
   ```bash
   # pooladm -c
   ```

3. (Optional) Copy the dynamic configuration to a static configuration file, for example, `/tmp/backup`.
   ```bash
   # 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 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. Test the validity of the configuration before committing it.
   ```bash
   # 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

More Information

- **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 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 113. 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 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 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 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 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 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 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 ($$) to pass the process number of the parent shell to the command.
   $ poolbind -q $$
   6384 pool backstage

   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 154 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.

```
machine% poolstat
   id  pool           pset
     0 pool_default  4 3.6 6.2
     1 pool_sales    4 3.3 8.4
```
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>2</td>
<td>165K</td>
<td>2.2</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>2</td>
<td>165K</td>
<td>2.2</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>2</td>
<td>10K</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 181
- “Consolidation Configuration” on page 182
- “Creating the Configuration” on page 182
- “Viewing the Configuration” on page 183

### 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.

- The application server has a two-CPU processor set.
- The database instance for the application server and the standalone database instance are consolidated onto a single processor set of at least four CPUs. The standalone 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)
devlopment: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)
```

Create an input file named `pool.host`, which will be used to configure the required resource pools. View the file.

```
# 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 app_pset (uint pset.min = 1; uint pset.max = 2)
create pset db_pset (uint pset.min = 4; uint pset.max = 6)
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 db_pset (string pset.poold.objectives="locality tight; utilization < 80")
modify pset app_pset (string pset.poold.objectives="locality tight; utilization < 80")
```

Update the configuration using the `pool.host` input file.

```
# poolcfg -f pool.host
```

Make the configuration active.

```
# pooladm -c
```

The framework is now functional on the system.

### Viewing the Configuration

To view the framework configuration, which also contains default elements created by the system, type:

```
# 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  pset_default

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 8
  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
Viewing the Configuration

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.pool.objectives locality tight; utilization < 80

pset pset_default
  int pset.sys_id -1
  string pset.units population
  boolean pset.default true
  uint pset.min 1
  uint pset.max 4294967295
  string pset.comment
  boolean pset.escapable false
  uint pset.load 0
  uint pset.size 0

cpu
  int cpu.sys_id 0
A graphic representation of the framework follows.

![Diagram of resource management configuration example](image)

**FIGURE 14-1** Server Consolidation Configuration

**Note** – In the pool `db_pool`, the standalone database instance is guaranteed 75 percent of the CPU resource.
This chapter describes the resource control and performance monitoring features in the Solaris Management Console. Only a subset of the resource management features can be controlled using the console.

You can use the console to monitor system performance and to enter the resource control values shown in Table 15–1 for projects, tasks, and processes. The console provides a convenient, secure alternative to the command-line interface (CLI) for managing hundreds of configuration parameters that are spread across many systems. Each system is managed individually. The console’s graphical interface supports all experience levels.

The following topics are covered.

- "Using the Console (Task Map)" on page 190
- "Console Overview" on page 190
- "Management Scope" on page 190
- "Performance Tool" on page 191
- "Resource Controls Tab" on page 194
- "Console References" on page 197
Using the Console (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the console</td>
<td>Start the Solaris Management Console in a local environment or in a name service or directory service environment. Note that the performance tool is not available in a name service environment.</td>
<td>“Starting the Solaris Management Console” in System Administration Guide: Basic Administration and “Using the Solaris Management Tools in a Name Service Environment (Task Map)” in System Administration Guide: Basic Administration</td>
</tr>
<tr>
<td>Monitor system performance</td>
<td>Access the Performance tool under System Status.</td>
<td>&quot;How to Access the Performance Tool” on page 191</td>
</tr>
<tr>
<td>Add resource controls to projects</td>
<td>Access the Resource Controls tab under System Configuration.</td>
<td>“How to Access the Resource Controls Tab” on page 195</td>
</tr>
</tbody>
</table>

Console Overview

Resource management functionality is a component of the Solaris Management Console. The console is a container for GUI-based administrative tools that are stored in collections called toolboxes. For information on the console and how to use it, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in System Administration Guide: Basic Administration.

When you use the console and its tools, the main source of documentation is the online help system in the console itself. For a description of the documentation available in the online help, see "Solaris Management Console (Overview)” in System Administration Guide: Basic Administration.

Management Scope

The term management scope refers to the name service environment that you choose to use with the selected management tool. The management scope choices for the resource control and performance tools are the /etc/project local file, or NIS.

The management scope that you select during a console session should correspond to the primary name service that is identified in the /etc/nsswitch.conf file.
Performance Tool

The Performance tool is used to monitor resource utilization. Resource utilization can be summarized for the system, viewed by project, or viewed for an individual user.

![Performance Tool in the Solaris Management Console](image)

**How to Access the Performance Tool**

The Performance tool is located under System Status in the Navigation pane. To access the Performance tool, do the following:

1. **Click the System Status control entity in the Navigation pane.**  
   The control entity is used to expand menu items in the Navigation pane.

2. **Click the Performance control entity.**

3. **Click the System control entity.**

4. **Double-click Summary, Projects, or Users.**  
   Your choice depends on the usage you want to monitor.
Monitoring by System

Values are shown for the following attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Processes</td>
<td>Number of processes that are active on the system</td>
</tr>
<tr>
<td>Physical Memory Used</td>
<td>Amount of system memory that is in use</td>
</tr>
<tr>
<td>Physical Memory Free</td>
<td>Amount of system memory that is available</td>
</tr>
<tr>
<td>Swap Used</td>
<td>Amount of system swap space that is in use</td>
</tr>
<tr>
<td>Swap Free</td>
<td>Amount of free system swap space</td>
</tr>
<tr>
<td>Page Rate</td>
<td>Rate of system paging activity</td>
</tr>
<tr>
<td>System Calls</td>
<td>Number of system calls per second</td>
</tr>
<tr>
<td>Network Packets</td>
<td>Number of network packets that are transmitted per second</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>Percentage of CPU that is currently in use</td>
</tr>
<tr>
<td>Load Average</td>
<td>Number of processes in the system run queue which are averaged over the last 1, 5, and 15 minutes</td>
</tr>
</tbody>
</table>

Monitoring by Project or User Name

Values are shown for the following attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Short Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Blocks</td>
<td>inblk</td>
<td>Number of blocks read</td>
</tr>
<tr>
<td>Blocks Written</td>
<td>oublk</td>
<td>Number of blocks written</td>
</tr>
<tr>
<td>Chars Read/Written</td>
<td>ioch</td>
<td>Number of characters read and written</td>
</tr>
<tr>
<td>Data Page Fault Sleep Time</td>
<td>dfmtime</td>
<td>Amount of time spent processing data page faults</td>
</tr>
<tr>
<td>Involuntary Context Switches</td>
<td>ictx</td>
<td>Number of involuntary context switches</td>
</tr>
<tr>
<td>System Mode Time</td>
<td>stime</td>
<td>Amount of time spent in the kernel mode</td>
</tr>
<tr>
<td>Attribute</td>
<td>Short Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>Major Page Faults</td>
<td>majfl</td>
<td>Number of major page faults</td>
</tr>
<tr>
<td>Messages Received</td>
<td>mrcv</td>
<td>Number of messages received</td>
</tr>
<tr>
<td>Messages Sent</td>
<td>msend</td>
<td>Number of messages sent</td>
</tr>
<tr>
<td>Minor Page Faults</td>
<td>minf</td>
<td>Number of minor page faults</td>
</tr>
<tr>
<td>Num Processes</td>
<td>nprocs</td>
<td>Number of processes owned by the user or the project</td>
</tr>
<tr>
<td>Num LWPs</td>
<td>count</td>
<td>Number of lightweight processes</td>
</tr>
<tr>
<td>Other Sleep Time</td>
<td>slptime</td>
<td>Sleep time other than tftime, dftime, kftime, and ltime</td>
</tr>
<tr>
<td>CPU Time</td>
<td>pctcpu</td>
<td>Percentage of recent CPU time used by the process, the user, or the project</td>
</tr>
<tr>
<td>Memory Used</td>
<td>pctmem</td>
<td>Percentage of system memory used by the process, the user, or the project</td>
</tr>
<tr>
<td>Heap Size</td>
<td>brksize</td>
<td>Amount of memory allocated for the process data segment</td>
</tr>
<tr>
<td>Resident Set Size</td>
<td>rsssize</td>
<td>Current amount of memory claimed by the process</td>
</tr>
<tr>
<td>Process Image Size</td>
<td>size</td>
<td>Size of the process image in Kbytes</td>
</tr>
<tr>
<td>Signals Received</td>
<td>sigs</td>
<td>Number of signals received</td>
</tr>
<tr>
<td>Stopped Time</td>
<td>stoptime</td>
<td>Amount of time spent in the stopped state</td>
</tr>
<tr>
<td>Swap Operations</td>
<td>swaps</td>
<td>Number of swap operations in progress</td>
</tr>
<tr>
<td>System Calls Made</td>
<td>sysc</td>
<td>Number of system calls made over the last time interval</td>
</tr>
<tr>
<td>System Page Fault Sleep Time</td>
<td>kftime</td>
<td>Amount of time spent processing page faults</td>
</tr>
<tr>
<td>System Trap Time</td>
<td>ttime</td>
<td>Amount of time spent processing system traps</td>
</tr>
<tr>
<td>Text Page Fault Sleep Time</td>
<td>tftime</td>
<td>Amount of time spent processing text page faults</td>
</tr>
<tr>
<td>Attribute</td>
<td>Short Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>User Lock Wait Sleep Time</td>
<td>ltime</td>
<td>Amount of time spent waiting for user locks</td>
</tr>
<tr>
<td>User Mode Time</td>
<td>utime</td>
<td>Amount of time spent in the user mode</td>
</tr>
<tr>
<td>User and System Mode Time</td>
<td>time</td>
<td>The cumulative CPU execution time</td>
</tr>
<tr>
<td>Voluntary Context Switches</td>
<td>vctx</td>
<td>Number of voluntary context switches</td>
</tr>
<tr>
<td>Wait CPU Time</td>
<td>wtime</td>
<td>Amount of time spent waiting for CPU (latency)</td>
</tr>
</tbody>
</table>

**Resource Controls Tab**

Resource controls allow you to associate a project with a set of resource constraints. These constraints determine the allowable resource usage of tasks and processes that run in the context of the project.
How to Access the Resource Controls Tab

The Resource Controls tab is located under System Configuration in the Navigation pane. To access Resource Controls, do the following:

1. Click the System Configuration control entity in the Navigation pane.
2. Double-click Projects.
3. Click on a project in the console main window to select it.
4. Select Properties from the Action menu.
5. Click the Resource Controls tab.

View, add, edit, or delete resource control values for processes, projects, and tasks.
Resource Controls You Can Set

The following table shows the resource controls that can be set in the console. 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 15-1  Standard Resource Controls Available in the Solaris Management Console

<table>
<thead>
<tr>
<th>Control Name</th>
<th>Description</th>
<th>Default Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>project.cpu-shares</td>
<td>The number of CPU shares that are granted to this project for use with the fair share scheduler (FSS) (see the FSS(7) man page)</td>
<td>Quantity (shares)</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 LWPs simultaneously available to this task's processes</td>
<td>Quantity (LWPs)</td>
</tr>
<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 that is available to this process</td>
<td>Index (maximum file descriptor)</td>
</tr>
<tr>
<td>process.max-file-size</td>
<td>Maximum file offset that is 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 that is created by this process</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td>process.max-data-size</td>
<td>Maximum heap memory that is available to this process</td>
<td>Size (bytes)</td>
</tr>
<tr>
<td>process.max-stack-size</td>
<td>Maximum stack memory segment that is 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, available to this process</td>
<td>Size (bytes)</td>
</tr>
</tbody>
</table>
Setting Values

You can view, add, edit, or delete resource control values for processes, projects, and tasks. These operations are performed through dialog boxes in the console.

Resource controls and values are viewed in tables in the console. The Resource Control column lists the resource controls that can be set. The Value column displays the properties that are associated with each resource control. In the table, these values are enclosed in parentheses, and they appear as plain text separated by commas. The values in parentheses comprise an “action clause.” Each action clause is composed of a threshold, a privilege level, one signal, and one local action that is associated with the particular threshold. Each resource control can have multiple action clauses, which are also separated by commas.

Note – On a running system, values that are altered in the project database through the console only take effect for new tasks that are started in a project.

Console References

For information on projects and tasks, see Chapter 2. For information on resource controls, see Chapter 6. For information on the fair share scheduler (FSS), see Chapter 8.

Note – Not all resource controls can be set in the console. See Table 15–1 for the list of controls that can be set in the console.
This part introduces 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.
Introduction to Solaris 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 covers the following 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 17.

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 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. The sparse root zone model optimizes the sharing of objects. The whole root zone model provides the maximum configurability. These concepts are discussed in Chapter 18.

**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 underutilized system. With the use of zones, these applications can be consolidated with other applications.

Zones allow you to delegate some administrative functions while maintaining overall system security.
How Zones Work

A non-global zone can be thought of as a box. One or more applications can run in this box without interacting with the rest of the system. 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.

Each zone that requires network connectivity has one or more dedicated IP addresses. 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.

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, 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 216.

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 317.

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 216.

The scheduling class for a non-global zone is set to the scheduling class for the system.
You can also set the scheduling class for a zone through the dynamic resource pools facility. If the zone is associated with a pool that has its pool.schedueler property set to a valid scheduling class, then processes running in the zone run in that scheduling class by default. To associate a zone with a resource pool, see Step 6 in “How to Configure the Zone” on page 234. For information on assigning a scheduling class to a resource pool, see “Introduction to Resource Pools” on page 136 and “How to Associate a Pool With a Scheduling Class” on page 167.

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.

**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>
</table>
| Global       | ■ Is assigned ID 0 by the system  
■ Provides the single instance of the Solaris kernel that is bootable and running on the system  
■ Contains a complete installation of the Solaris system software packages  
■ Can contain additional software packages or additional software, directories, files, and other data not installed through packages  
■ Provides a complete and consistent product database that contains information about all software components installed in the global zone  
■ Holds configuration information specific to the global zone only, such as the global zone host name and file system table  
■ Is the only zone that is aware of all devices and all file systems  
■ Is the only zone with knowledge of non-global zone existence and configuration  
■ Is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled |
### How Zones Work

<table>
<thead>
<tr>
<th>Type of Zone</th>
<th>Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Global</td>
<td>- Is assigned a zone ID by the system when the zone is booted</td>
</tr>
<tr>
<td></td>
<td>- Shares operation under the Solaris kernel booted from the global zone</td>
</tr>
<tr>
<td></td>
<td>- Contains an installed subset of the complete Solaris Operating System software packages</td>
</tr>
<tr>
<td></td>
<td>- Contains Solaris software packages shared from the global zone</td>
</tr>
<tr>
<td></td>
<td>- Can contain additional installed software packages not shared from the global zone</td>
</tr>
<tr>
<td></td>
<td>- Can contain additional software, directories, files, and other data created on the non-global zone that are not installed through packages or shared from the global zone</td>
</tr>
<tr>
<td></td>
<td>- Has a complete and consistent product database that contains information about all software components installed on the zone, whether present on the non-global zone or shared read-only from the global zone</td>
</tr>
<tr>
<td></td>
<td>- Is not aware of the existence of any other zones</td>
</tr>
<tr>
<td></td>
<td>- Cannot install, manage, or uninstall other zones, including itself</td>
</tr>
<tr>
<td></td>
<td>- Has configuration information specific to that non-global zone only, such as the non-global zone host name and file system table</td>
</tr>
</tbody>
</table>

## How Non-Global Zones Are Administered

A global administrator has 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.

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 global administrator can log in to the installed zone by using the zlogin command. At first login, the internal configuration for the zone is completed. The zoneadm command is then used to boot the zone.
For information about zone configuration, see Chapter 17. For information about zone installation, see Chapter 19. For information about zone login, see Chapter 21.

**Non-Global Zone State Model**

A non-global zone can be in one of the following six states:

- **Configured** The zone's configuration is complete and committed to stable storage. However, those elements of the zone's application environment that must be specified after initial boot are not yet present.

- **Incomplete** During an install or uninstall operation, `zoneadm` sets the state of the target zone to incomplete. Upon successful completion of the operation, the state is set to the correct state.

- **Installed** The zone's configuration is instantiated on the system. The `zoneadm` command is used to verify that the configuration can be successfully used on the designated Solaris system. Packages are installed under the zone's root path. In this state, the zone has no associated virtual platform.

- **Ready** The virtual platform for the zone is established. The kernel creates the `zsched` process, network interfaces are plumbed, file systems are mounted, and devices are configured. A unique zone ID is assigned by the system. At this stage, no processes associated with the zone have been started.

- **Running** User processes associated with the zone application environment are running. The zone enters the running state as soon as the first user process associated with the application environment (`init`) is created.

- **Shutting down and Down** These states are transitional states that are visible while the zone is being halted. However, a zone that is unable to shut down for any reason will stop in one of these states.

Chapter 20 and the `zoneadm(1M)` man page describe how to use the `zoneadm` command to initiate transitions between these states.
### TABLE 16-1  Commands That Affect Zone State

<table>
<thead>
<tr>
<th>Current Zone State</th>
<th>Applicable Commands</th>
</tr>
</thead>
</table>
| Configured         | `zonecfg -z zonename verify`  
                        | `zonecfg -z zonename commit`  
                        | `zonecfg -z zonename delete`  
                        | `zoneadm -z zonename attach`  
                        | `zoneadm -z zonename verify`  
                        | `zoneadm -z zonename install`  |
| Incomplete         | `zoneadm -z zonename uninstall`  |
| Installed          | `zoneadm -z zonename ready` (optional)  
                        | `zoneadm -z zonename boot`  
                        | `zoneadm -z zonename uninstall` uninstalls the configuration of the specified zone from the system.  
                        | `zoneadm -z zonename move path`  
                        | `zoneadm -z zonename detach`  
                        | `zonecfg -z zonename` can be used to add or remove an fs, dataset, net, device, rctl, or attr property. The zonepath and inherit-pkg-dir resources cannot be changed.  |
| Ready              | `zoneadm -z zonename boot`  
                        | `zoneadm halt` and system reboot return a zone in the ready state to the installed state.  
                        | `zonecfg -z zonename` can be used to add or remove an fs, dataset, net, device, rctl, or attr property. The zonepath and inherit-pkg-dir resources cannot be changed.  |
| Running            | `zlogin options zonename`  
                        | `zoneadm -z zonename reboot`  
                        | `zoneadm -z zonename halt` returns a ready zone to the installed state.  
                        | `zoneadm halt` and system reboot return a zone in the running state to the installed state.  
                        | `zonecfg -z zonename` can be used to add or remove an fs, dataset, net, device, rctl, or attr property. The zonepath and inherit-pkg-dir resources cannot be changed.  |
Note – Parameters changed through `zonecfg` do not affect a running zone. The zone must be rebooted for the changes to take effect.

Non-Global Zone Characteristics

A zone provides isolation at almost any level of granularity you require. A zone does not need a dedicated CPU, a physical device, or a portion of physical memory. These resources can either be multiplexed across a number of zones running within a single domain or system, or allocated on a per-zone basis using the resource management features available in the operating system.

Each zone can provide a customized set of services. To enforce basic process isolation, a process can see or signal only those processes that exist in the same zone. Basic communication between zones is accomplished by giving each zone at least one logical network interface. An application running in one zone cannot observe the network traffic of another zone. This isolation is maintained even though the respective streams of packets travel through the same physical interface.

Each zone is given a portion of the file system hierarchy. Because each zone is confined to its subtree of the file system hierarchy, a workload running in a particular zone cannot access the on-disk data of another workload running in a different zone.

Files used by naming services reside within a zone’s own root file system view. Thus, naming services in different zones are isolated from one other and the services can be configured differently.

Using Resource Management Features With Non-Global Zones

If you use resource management features, you should align the boundaries of the resource management controls with those of the zones. This alignment creates a more complete model of a virtual machine, where namespace access, security isolation, and resource usage are all controlled.

Any special requirements for using the various resource management features with zones are addressed in the individual chapters of this manual that document those features.
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 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.

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>Determine how many zones to configure.</td>
<td>Assess:  ■ The performance requirements of the applications you intend to run in zones  ■ The availability of the recommended 100 MB of free disk space per zone to be installed</td>
<td>See “Evaluating the Current System Setup” on page 228.</td>
</tr>
<tr>
<td>Determine whether you plan to use resource management features, such as resource pools, with zones.</td>
<td>If you are also using resource management features on your system, align the zones with the resource management boundaries. If you are using resource pools, configure the pools if necessary.</td>
<td>See Chapter 1, Chapter 13</td>
</tr>
<tr>
<td>Perform the preconfiguration tasks.</td>
<td>Determine the zone name and the zone path, obtain IP addresses, and determine the required file systems and devices for each zone.  (Optional) To set a default scheduling class for the non-global zone that is different from the system default, you can associate the zone with a pool that specifies a default scheduler. The scheduling class is set at the pool level through the poolscheduler property.</td>
<td>For information on the zone name and path, IP addresses, file systems, and devices, see Chapter 17 and “Evaluating the Current System Setup” on page 228.  For information on resource pool association, see “How Zones Work” on page 204 and “How to Configure the Zone” on page 234.</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>Develop configurations.</td>
<td>Configure non-global zones.</td>
<td>See &quot;Configuring, Verifying, and Committing a Zone&quot; on page 233 and the zonecfg(1M) man page.</td>
</tr>
<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 19 and Chapter 20.</td>
</tr>
<tr>
<td>As global administrator, log in to each non-global zone using the zlogin command with the -C or place a sysidcfg file in the zone's /etc directory.</td>
<td></td>
<td>See Chapter 21 and Chapter 22.</td>
</tr>
<tr>
<td>As global administrator, boot the non-global zone.</td>
<td>Boot each zone to place the zone in the running state.</td>
<td>See Chapter 19 and Chapter 20.</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 the zones environment are covered in this guide.</td>
</tr>
</tbody>
</table>
This chapter provides an introduction to non-global zone configuration.

The following topics are covered in this chapter:
- “What’s New in This Chapter?” on page 213
- “Pre-Installation Configuration Process” on page 214
- “Zone Components” on page 214
- “Using the zonecfg Command” on page 216
- “zonecfg Modes” on page 217
- “Zone Configuration Data” on page 219
- “Tecla Command-Line Editing Library” on page 225

After you have learned about zone configuration, go to Chapter 18 to configure non-global zones for installation on your system.

What's New in This Chapter?

**Solaris 10 6/06:** Support for the Zettabyte File System (ZFS) in non-global zones has been added. See “Resource Type Properties” on page 223 for more information.

**Solaris 10 11/06:** Support for configurable privileges has been added. See “Configurable Privileges” on page 215.

For a complete listing of new Solaris 10 features and a description of Solaris releases, see *Solaris 10 What’s New.*
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 zone resources and properties that can be configured.

Zone Name and Path

You must choose a name and a path for your zone.

Resource Pool Association

If you plan to associate the zone with a resource pool other than the system default, configure resource pools as described in Chapter 13 before you configure the zone.

Zone Interfaces

Each zone that requires network connectivity must have one or more dedicated IP addresses. These addresses are associated with logical network interfaces. Zone interfaces configured by the `zonecfg` command will automatically be plumbed and placed in the zone when it is booted.

The `ifconfig` command can be used from the global zone to add or remove logical interfaces in a running zone. For more information, see “Network Interfaces” on page 325.

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 317 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 234.

**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, regardless of project. These limits are specified in the zonecfg configuration. For more information, see "How to Configure the Zone" on page 234.

**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.
Using the zonecfg Command

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 330, “How to Configure the Zone” on page 234, 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 234.

Using the zonecfg Command

The zonecfg command, which is described in the zonecfg(1M) man page, is used to configure a zone. 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
- 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 18.
**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.

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

- `add` In the global scope, add the specified resource type to the configuration.
  
  In the resource scope, add a property of the given name with the given value.
See “How to Configure the Zone” on page 234 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.

**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.

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.

**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 217 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:

**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 alpha-numeric character.

  The name can contain alpha-numeric 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.
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</td>
<td>zonecfg zonepath</td>
</tr>
<tr>
<td>/home/export/my-zone/root</td>
<td>Root of the zone</td>
</tr>
<tr>
<td>/home/export/my-zone/dev</td>
<td>Devices created for the zone</td>
</tr>
</tbody>
</table>

See “Traversing File Systems” on page 323 for a further discussion of this issue.

**Note** – See “Solaris 10 6/06 and Solaris 10 11/06: Do Not Place the Root File System of a Non-Global Zone on ZFS” on page 370 for ZFS restrictions for this release.

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 autoboott, 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
```

This property is used to associate the zone with a resource pool on the system. Multiple zones can share the resources of one pool.

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 317.

**dataset**

Adding a ZFS file system 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 29 for information on dataset issues.

**inherit-pkg-dir**

This resource should not be configured in a whole root zone.

In a 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`
The network interface resource is the virtual interface name. Each zone can have network interfaces that should be plumbed when the zone transitions from the installed state to the ready state.

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.

The rctl resource is used for zone-wide resource controls. The controls should be enabled when the zone transitions from the installed state to the ready state. The zone-wide resource controls implemented in this release are zone.cpu-shares and zone.max-lwps.

This property is used to specify a privilege mask other than the default. See "Privileges in a Non-Global Zone" on page 330.

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.

This generic attribute can be used for user comments or by other subsystems. The name property of an attr must begin with an alpha-numeric character. The name property can contain alpha-numeric characters, hyphens (-), and periods (.). Attribute names beginning with zone. are reserved for use by the system.
Resource Type Properties

Some resource types also have properties to configure. The following properties are associated with the resource types shown.

**fs**

- dir, special, raw, type, options

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 `nodevices` and `logging` are added.

```
zonecfg:my-zone> add fs
zonecfg:my-zone:fs> set dir=/mnt
zonecfg:my-zone:fs> set special=/dev/dsk/c0t0d0s2
zonecfg:my-zone:fs> set raw=/dev/rdsk/c0t0d0s2
zonecfg:my-zone:fs> set type=ufs
zonecfg:my-zone:fs> add options [nodevices, logging]
zonecfg:my-zone:fs> end
```

For more information, see "The `-o nosuid` Option" on page 317, "Security Restrictions and File System Behavior" on page 320, 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**

- name

The lines in the following example specify that the dataset `sales` is to be visible and mounted in the non-global zone and no longer visible in the global zone.

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

---

**inherit-pkg-dir**

- 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
```

In the following example, IP address 192.168.0.1 is added to a zone. An hme0 card is used for the physical interface.

```
zonecfg:my-zone> add net
zonecfg:my-zone:net> set physical=hme0
zonecfg:my-zone:net> set address=192.168.0.1
zonecfg:my-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.

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
```

In this release, there are two zone-wide resource controls, `zone.cpu-shares` and `zone.max-lwps`.

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 359.

The `zone.max-lwps` resource control enhances resource isolation by preventing too many LWPs in one zone from affecting other zones. A
zone's total LWPs can be further subdivided among projects within the zone by using project.max-lwps entries.

Note that zone-wide resource control entries in a zone are configured differently 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.

For general information about resource controls and attributes, see Chapter 6 and "Resource Controls Used in Non-Global Zones" on page 328.

attr name, type, value

In the following example, a comment about a zone is added.

You can use the export subcommand to print a zone configuration to standard output. The configuration is saved in a form that can be used in a command file.

Tecla Command-Line Editing Library

The Tecla command-line editing library is included for use with the zonecfg command. The library provides a mechanism for command-line history and editing support.

The Tecla command-line editing library is documented in the following man pages:

- enhance(1)
- libtecla(3LIB)
Tecla Command-Line Editing Library

- ef_expand_file(3TECLA)
- gl_get_line(3TECLA)
- gl_io_mode(3TECLA)
- pca_lookup_file(3TECLA)
- tecla(5)
Planning and Configuring Non-Global Zones (Tasks)

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 17.

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. | Refer to historical usage. Also see “Disk Space Requirements” on page 228 and “Resource Pools Used in Zones” on page 137. |
### Evaluating the Current System Setup

Zones can be used on any machine that runs the Solaris 10 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.

---

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine the name for the zone.</td>
<td>Decide what to call the zone based on the naming conventions.</td>
<td>See &quot;Zone Configuration Data&quot; on page 219 and &quot;Zone Host Name&quot; on page 230.</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 &quot;Determine the Zone Host Name and Obtain the Network Address&quot; on page 230 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 &quot;File Systems Mounted in Zones&quot; on page 214 for more information.</td>
</tr>
<tr>
<td>Determine which network interfaces should be plumbed in the zone.</td>
<td>Review your application requirements.</td>
<td>See &quot;Network Interfaces&quot; on page 325 for more information.</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>Determine the zone path.</td>
<td>Each zone has a path to its root directory that is relative to the global zone’s root directory.</td>
<td>See &quot;Zone Configuration Data&quot; on page 219.</td>
</tr>
<tr>
<td>Configure the zone.</td>
<td>Use zonecfg to create a configuration for the zone.</td>
<td>See &quot;Configuring, Verifying, and Committing a Zone&quot; on page 233.</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 &quot;Configuring, Verifying, and Committing a Zone&quot; on page 233.</td>
</tr>
</tbody>
</table>
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.

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.
- 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.
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 234.

**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 Obtain the Network Address**

You must determine the host name for the zone. Then, 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.

**Zone Host Name**

The host name you select for the zone must be defined either in the `hosts` database or in the `/etc/inet/ipnodes` 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 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.


Zone Network Address

Each 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 ifconfig command with the -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 fe80::c0a8:c80a:1/10. If the global prefix in use on that subnet is 2001:0db8:aa:ccdd/64, a unique global unicast address for the same non-global zone is 2001:0db8:aa: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.

**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) virtual 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 317.

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.

See the \texttt{lofs(7S)} man page for more information.
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 a non-global zone’s configuration.</td>
<td>“Configuring, Verifying, and Committing a Zone” on page 233, “Script to Configure Multiple Zones” on page 238</td>
</tr>
<tr>
<td>Modify a zone configuration.</td>
<td>Use this procedure to modify a resource type in a zone configuration or add a dedicated device to a zone.</td>
<td>“Using the <code>zonecfg</code> Command to Modify a Zone Configuration” on page 241</td>
</tr>
<tr>
<td>Revert a zone configuration or delete a zone configuration.</td>
<td>Use the <code>zonecfg</code> command to undo a resource setting made to a zone configuration or to delete a zone configuration.</td>
<td>“Using the <code>zonecfg</code> Command to Revert or Remove a Zone Configuration” on page 243</td>
</tr>
<tr>
<td>Delete a zone configuration.</td>
<td>Use the <code>zonecfg</code> command with the delete subcommand to delete a zone configuration from the system.</td>
<td>“How to Delete a Zone Configuration” on page 245</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

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 243.

A script to configure multiple zones on your system is provided in “Script to Configure Multiple Zones” on page 238.
To display a non-global zone's configuration, see “How to Display the Configuration of a Non-Global Zone” on page 241.

▶ How to Configure 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 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
   ```
   
   Do not place the zonepath on ZFS for this release.

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:my-zone> set autoboot=true
   ```

6 If resource pools are enabled on your system, associate a pool with the zone.
   This example uses the default pool, named pool_default.

   ```
   zonecfg:my-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 167 and “Creating the Configuration” on page 182.
7 Revise the default set of privileges.

```
zonecfg:my-zone> set limitpriv="default,proc_lock_memory"
```

This line adds the ability for a process to lock pages in physical memory to the default set of privileges.

8 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.

9 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
```
10  **(Sparse Root Zone Only) Add a shared file system 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 228.

```
zonecfg:my-zone> add inherit-pkg-dir
```

- **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.

- **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`

11  **Add a network virtual interface.**

```
zeonefg:my-zone> add net
```

- **Set the IP address for the network interface, 192.168.0.1 in this procedure.**
  
  ```
zeonefg:my-zone:net> set address=192.168.0.1
  ```

- **Set the physical device type for the network interface, the hme device in this procedure.**
  
  ```
zeonefg:my-zone:net> set physical=hme0
  ```

- **End the specification.**

  ```
zeonefg:my-zone:net> end
  ```

This step can be performed more than once to add more than one network interface.
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.
```
zonecfg:my-zone> add rctl
```  
   a. Set the name of the resource control, zone.cpu-shares in this procedure.
```
zonecfg:my-zone:rctl> set name=zone.cpu-shares
```  
   b. Add values for the privilege, the share limit, and the action to be taken when that threshold is reached.
```
zonecfg:my-zone:rctl> add value (priv=privileged, limit=20, action=none)
```  
   c. End the rctl specification.
```
zonecfg:my-zone:rctl> end
```  
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.
```
zeonecfg: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.

More Information

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 254 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.

#!/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>"

16 Commit the zone configuration for the zone.

zonecfg:my-zone> commit

17 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.

More Information

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 254 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.

#!/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
        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
            fi
        fi
        j='expr $j + 1'
    done
    i='expr $i + 1'
done
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/wzkSbc" >> $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# zonecfg -z zonename info
   ```

---

**Using the zonecfg Command to Modify a Zone Configuration**

You can also use the zonecfg command to do the following:

- Modify a resource type 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)
   ```
Add the new value.
```console
zonecfg:my-zone: rctl> add value (priv=privileged, limit=10, action=none)
```

End the revised rctl specification.
```console
zonecfg:my-zone: rctl> end
```

Commit the zone configuration for the zone.
```console
zonecfg:my-zone> commit
```

Exit the zonecfg command.
```console
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 Modify a Property Type in a Zone Configuration

Use this procedure to reset a standalone property that does not have related properties to configure. For example, to remove the existing pool association, you can reset the pool resource to null.

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.**
   ```console
   global# zonecfg -z my-zone
   ```

3. **Reset the property to be changed, the existing pool association in this procedure.**
   ```console
   zonecfg:my-zone> set pool=""
   ```

4. **Commit the zone configuration for the zone.**
   ```console
   zonecfg:my-zone> commit
   ```

5. **Exit the zonecfg command.**
   ```console
   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 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:myzone> add device
   ```

3. **Set the device match, */dev/scsi/scanner/c3t4* in this procedure.**

   ```
   zonecfg:myzone:device> set match=/dev/scsi/scanner/c3t4*
   ```

4. **End the device specification.**

   ```
   zonecfg:myzone:device> end
   ```

5. **Exit the zonecfg command.**

   ```
   zonecfg:myzone> 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, and uninstalling zones is also provided.

The following topics are addressed in this chapter:

- "Zone Installation Concepts" on page 247
- "Zone Construction" on page 248
- "The zoneadm Daemon" on page 249
- "The zsched Zone Scheduler" on page 250
- "Zone Application Environment" on page 250
- "About Halting, Rebooting, and Uninstalling Zones" on page 250

To install and boot a non-global zone, or to halt or uninstall a non-global zone, see Chapter 20.

**Zone Installation 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, which is similar to booting a regular Solaris system
- Display information about a running zone
- Halt a zone
- Reboot a zone
- Uninstall a zone

For zone installation and verification procedures, see Chapter 20 and the zoneadm(1M) man page. Also refer to the zoneadm(1M) man page for supported options to the zoneadm list
command. For zone configuration procedures, see Chapter 18 and the 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 334 before you install non-global zones.

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 open networking configuration (generic_open.xml). Network configuration types are described in Chapter 15, “Managing Services (Tasks),” in System Administration Guide: Basic Administration. The zone administrator can switch the zone to the limited networking configuration (generic_limited_net.xml) by using the netservices command. Specific services can be enabled or disabled by using SMF commands.

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

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, 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 334.

The following features cannot be configured in a non-global zone:
- Solaris Live Upgrade™ boot environments
- Solaris Volume Manager metadevices
- DHCP address assignment
- 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 plumbed, and devices are configured. Transitioning into the ready state prepares the virtual platform to begin running user processes. In the ready state, the zsched and zoneadm processes are started to manage the virtual platform.

- zsched, a system scheduling process similar to sched, 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 zoneadm Daemon

The zones administration daemon, zoneadm, 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 zoneadm process running for each active (ready, running, or shutting down) zone on the system.

The zoneadm 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. For more information, see the devfsadm(1M) man page.
- Plumbing virtual network interfaces.
- Mounting loopback and conventional file systems.
Instantiating and initializing the zone console device.

Unless the zoneadm 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 zoneadm.

The man page for the zoneadm daemon is zoneadm(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.

Before 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. The application environment is established by responding to a series of prompts that appear on the zone console, as explained in “Internal Zone Configuration” on page 266. 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, and uninstalling zones. Troubleshooting tips for zones that fail to halt when requested are also provided.

### 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 unplumbed, 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 276.
If the halt operation fails, see “Zone Does Not Halt” on page 371.

**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 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 the zones to autoboott, 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.

- 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.

See “Cloning a Non-Global Zone on the Same System” on page 262 for more information.
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 also addressed. The procedure to completely delete a zone from a system is also provided.

For general information about zone installation and related operations, see Chapter 19.

### 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><em>(Optional) How to Verify a Configured Zone Before It Is Installed</em> on page 254</td>
</tr>
<tr>
<td>Install a configured zone.</td>
<td>Install a zone that is in the configured state.</td>
<td><em>How to Install a Configured Zone</em> on page 255</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><em>(Optional) How to Transition the Installed Zone to the Ready State</em> on page 256</td>
</tr>
<tr>
<td>Boot a zone.</td>
<td>Booting a zone places the zone in the running state. A zone can be booted from the ready state or from the installed state. Note that you must perform the internal zone configuration before you boot the zone for the first time. This is described in <em>Internal Zone Configuration</em> on page 266.</td>
<td><em>How to Boot a Zone</em> on page 256, <em>Performing the Initial Internal Zone Configuration</em> on page 270</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 233.

((Optional) How to Verify a Configured Zone Before It Is Installed

You can verify a zone prior to installing it. 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

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 the configured zone** `my-zone` **by using the zoneadm command with the -z install option.**

   ```
   global# zoneadm -z my-zone install
   ```

   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.

4 **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:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>- my-zone</td>
<td>installed</td>
<td>/export/home/my-zone</td>
</tr>
</tbody>
</table>

**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 open network configuration described in Chapter 15, “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 276 for details.

▼ **(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:

```
ID  NAME   STATUS   PATH
 0  global running  /
 1  my-zone ready  /export/home/my-zone
```

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 log in to the zone for the first time. This is described in “Internal Zone Configuration” on page 266.

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 272, 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:

   ID NAME STATUS PATH
   0 global running /
   1 my-zone running /export/home/my-zone

   If you see the following message when you boot the zone:

   # 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.

▼ 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 21 and Chapter 22.

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 “How to Use zlogin to Shut Down a Zone” on page 276.</td>
<td>“How to Halt a Zone” on page 259</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 260</td>
</tr>
</tbody>
</table>
### Halting, Rebooting, and Uninstalling Zones

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uninstall a zone.</td>
<td>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 261</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>“Cloning a Non-Global Zone on the Same System” on page 262</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>“Deleting a Non-Global Zone From the System” on page 263</td>
</tr>
</tbody>
</table>

#### 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 276.

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>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
</tr>
<tr>
<td>1</td>
<td>my-zone</td>
<td>running</td>
<td>/export/home/my-zone</td>
</tr>
</tbody>
</table>

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
   ```
Halting, Rebooting, and Uninstalling Zones

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>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>- my-zone</td>
<td>installed</td>
<td>/export/home/my-zone</td>
</tr>
</tbody>
</table>

5 Boot the zone if you want to restart it.

    global# zoneadm -z my-zone boot

Troubleshooting If the halt operation fails, see “Zone Does Not Halt” on page 371 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>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
</tr>
<tr>
<td>1</td>
<td>my-zone</td>
<td>running</td>
<td>/export/home/my-zone</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>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
</tr>
<tr>
<td>2</td>
<td>my-zone</td>
<td>running</td>
<td>/export/home/my-zone</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:

   ID NAME      STATUS     PATH
   0 global    running    /
   - my-zone   installed  /export/home/my-zone

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

4 List the zones on the system again, to verify that my-zone is no longer listed.

   global# zoneadm list -v

   You will see a display that is similar to the following:

   ID NAME      STATUS     PATH
   0 global    running    /

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 zone path to a target zone path.

▼ 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 234 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 zone path for the new zone, but you can change other attributes, such as the IP address, as well.**

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 zone path /export/home/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
   ```

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
   ```

4. 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>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>my-zone</td>
<td></td>
<td>/export/home/my-zone</td>
</tr>
<tr>
<td></td>
<td>zonel</td>
<td></td>
<td>/export/home/zonel</td>
</tr>
</tbody>
</table>

   List the zones on the system.
This chapter discusses logging in to zones from the global zone.

The following topics are covered in this chapter:

- “zlogin Command” on page 265
- “Non-Global Zone Login Methods” on page 266
- “Interactive and Non-Interactive Modes” on page 267
- “Failsafe Mode” on page 267
- “Remote Login” on page 267

For procedures and usage information, see Chapter 22.

**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 275, 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.
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 zone console login is used. 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 270.

- 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. 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 267.

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 user name, 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, logins through `rlogin`, `ssh`, and `telnet` function normally. For more information about these 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 21.

**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>“Performing the Initial Internal Zone Configuration” on page 270</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>“Logging In to a Zone” on page 273</td>
</tr>
<tr>
<td>Exit a non-global zone.</td>
<td>Disconnect from a non-global zone.</td>
<td>“How to Exit a Non-Global Zone” on page 275</td>
</tr>
<tr>
<td>Shut down a zone.</td>
<td>Shut down a zone by using the shutdown utility or a script.</td>
<td>“How to Use zlogin to Shut Down a Zone” on page 276</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 266.
- 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 272.

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 364 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:
Select a Language
   1. English
   2. es
   2. fr
Please make a choice (0 - 1), 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.ISO8859-1)
   7. Go Back to Previous Screen
Please make a choice (0 - 9), 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 the complete list of questions you must answer, see "Internal Zone Configuration" on page 266.
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:

```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)
}
timezone=US/Central
root_password=m4qtoWN
```
4 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:

```bash
global# vi default/nfs
```
```
.
.
.
   NFSMAPID_DOMAIN=domain
```

For more information on the NFSv4 domain parameter, see the `nfsmapid(1M)` man page.

5 Create the file `.NFS4inst_state.domain` in this directory to indicate that the NFSv4 domain has been set:

```bash
global# touch .NFS4inst_state.domain
```

6 Boot the zone.

---

## 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 267.

### ▼ 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`.**
   ```bash
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:

```text
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.
```

3. **When the zone console displays, log in as root, press Return, and type the root password when prompted.**

   ```text
   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.**

   ```text
   global# zlogin my-zone
   ```

   Information similar to the following will display:

   ```text
   [Connected to zone 'my-zone' pts/2] 
   Last login: Wed Jul  3 16:25:00 on console 
   Sun Microsystems Inc. SunOS 5.10 Generic June 2004
   ```

3. **Type exit to close the connection.**

   You will see a message similar to the following:

   ```text
   [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*.

2. **From the global zone, log in to the my-zone zone and supply a command name.**
   The command zonename is used here.
   
   ```
   zonename# zlogin my-zone zonename
   ```
   
   You will see the following output:
   
   ```
   my-zone
   ```

How to Exit a Non-Global Zone

- **To exit the zone, type:**
  
  ```
  zonename# exit
  ```
  
  Your screen will look similar to this:
  
  ```
  [Connection to zone 'my-zone' pts/6 closed]
  ```
  
  You can also disconnect from a non-global zone by typing the tilde (~) character and a period.
  
  ```
  zonename# ~.
  ```

See Also

For more information about zlogin command options, see zlogin(1).

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 259.

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 -i 0
```

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.

---

**Switching the Non-Global Zone to a Different Networking Service Configuration**

This zone was installed with the open networking configuration described in Chapter 15, “Managing Services (Tasks),” in *System Administration Guide: Basic Administration*. You can switch the zone to the limited networking configuration, or enable or disable individual services in the zone.
How to Switch the Zone to the Limited 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 limited networking configuration.
   my-zone# /usr/sbin/netservices limited
   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 physical memory control using the resource capping daemon.
   my-zone# svcadm enable svc:/system/rcap:default

3. List the services to verify that rcapd is enabled.
   my-zone# svcs -a

   online 14:04:21 svc:/system/rcap:default

   .
   .
   .

Switching the Non-Global Zone to a Different Networking Service Configuration
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 is new for the Solaris 10 11/06 release.

This chapter describes how to:
- Move an existing non-global zone to a new location on the same machine
- Migrate an existing non-global zone to a new machine

**Solaris 10 11/06: 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 219 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.**
   ```shell
   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.**
   ```shell
   global# zoneadm -z db-zone move /export/zones/db-zone
   ```
Solaris 10 11/06: Migrating a Non-Global Zone to a Different Machine

The `zoneadm` command can be used to migrate an existing non-global zone from one system to another.

About Migrating a Zone

The zone is halted and detached from its current host. The `zonepath` is moved to the target host, where it is attached.

The following restrictions 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 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.

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.

Verify the path.

```
global# zoneadm list -iv
   ID   NAME STATUS PATH
     0 global running /
       - my-zone installed /export/home/my-zone
       - db-zone installed /export/zones/db-zone
```
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.**
   ```bash
   host1# zoneadm -z my-zone halt
   ```

3. **Detach the zone.**
   ```bash
   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 282 for more information.

5. **On the new host, configure the zone.**
   ```bash
   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.**
   ```bash
   zonecfg:my-zone> create -a /export/zones/my-zone
   ```

7. **(Optional) View the configuration.**
   ```bash
   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 (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: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.

- 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.

- 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 23–1  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.

   ```
   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.

   ```
   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 372 for troubleshooting information on the following:
- Patches and packages are out of sync.
- Operating system releases do not match.
About Packages and Patches on a Solaris System With Zones Installed (Overview)

Solaris 10 1/06: This chapter has been completely revised.

This chapter discusses maintaining the Solaris Operating System when zones are installed. Information about adding packages and patches to the operating system 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 10 Release and Installation Collection and System Administration Guide: Basic Administration for more information.

This chapter covers the following topics:

- “What’s New in Packaging and Patching When Zones Are Installed” on page 286
- “Packaging and Patch Tools Overview” on page 286
- "About Packages and Zones" on page 287
- "Keeping Zones in Sync" on page 288
- “About Adding Packages in Zones” on page 289
- “About Removing Packages in Zones” on page 292
- “Package Parameter Information” on page 294
- "Package Information Query" on page 301
- “About Adding Patches in Zones” on page 301
- “Applying Patches on a Solaris System With Zones Installed” on page 302
- “Removing Patches on a Solaris System With Zones Installed” on page 303
- “PatchPro Support” on page 304
- "Product Database" on page 304
What's New in Packaging and Patching When Zones Are Installed

Solaris 10 1/06: This chapter has been rewritten since Solaris 10, to document the current behavior of the package and patch commands on a system with installed non-global zones.

Solaris 10 6/06: Information on the SUNW_PKG_ALLZONES, SUNW_PKG_HOLLOW, and SUNW_PKG_THISZONE package parameters has been revised. See “Packaging and Patch Tools Overview” on page 286 and “Package Parameter Information” on page 294.

Solaris 10 6/06 and later releases: The Sun Update Connection—System is now available as software update 121118 (SPARC) or 121119 (x86/x64). Sun Update Connection can now update a Solaris system with non-global zones installed when managed from the global zone. For more information, see Sun Update Connection, System Edition 1.0 Release Notes, Chapter 16, “Managing Software (Overview),” in System Administration Guide: Basic Administration, and Chapter 18, “Registering Your Solaris 10 6/06 or 1/06 Software (Tasks),” in System Administration Guide: Basic Administration.

For a complete listing of new Solaris 10 features and a description of Solaris releases, see Solaris 10 What’s New.

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.

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 297.
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 299.

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 300.

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 233 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 operations are performed, a zone is temporarily locked. The system will also confirm a requested operation with the administrator before proceeding.

---

**About 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" on page 294. See "Checking Package Parameter Settings on a System with Zones Installed" on page 313 for usage instructions.
For information about package characteristics and parameters, see the pkginfo(4) man page. For information about displaying package parameter values, see the pkgparam(1) man page.

**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 -G option. For more information about this option, see "About Adding Packages in Zones" on page 289.

**Keeping Zones in Sync**

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.

About Adding Packages in Zones

The pkgadd system utility described in the pkgadd(1M) man page is used to add packages on a Solaris system with zones installed.

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`. 
About Removing Packages in Zones

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

The pkg rm utility described in the pkg rm(1M) man page supports removing packages on a Solaris system with zones installed.

Using pkg rm in the Global Zone

The pkg rm 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 pkg rm utility is used in the global zone, the following actions apply.

- pkg rm 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.
- pkg rm 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 pkg rm utility in the global zone. As the global administrator, run pkg rm 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 `pkgrm` in a Non-Global Zone**

As the zone administrator, use the `pkgrm` utility in a non-global zone to remove a package. The following limitations apply:

- `pkgrm` can only remove packages from the non-global zone.
- The `-G` option cannot be used. If this option is used, `pkgrm` 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 `pkgrm` utility:

- Dependencies are checked on the non-global zone's package database. If the dependency check fails, then `pkgrm` 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.
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>
<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></td>
<td>In both cases, the entire contents of the package is visible in all zones where the package is installed.</td>
</tr>
</tbody>
</table>
### TABLE 24-1 Valid Package Parameter Settings (Continued)

<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>
</table>
| false                     | false                   | true                      | 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.  
  - If the `pkgadd` command is run in the global zone, the package is installed in the global zone only.  
  - If the `pkgadd` command is run in a non-global zone, the package is installed in the non-global zone only.  
  In both cases, the entire contents of the package is visible in the zone where the package is installed. |
| true                      | false                   | false                     | A package with these settings can be installed in the global zone only. When the `pkgadd` 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.  
  **Note** – Any attempt to install the package in a non-global zone fails. |
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 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.
SUNW_PKG_ALLZONES Package Parameter

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.
### 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>
<tbody>
<tr>
<td><code>false</code></td>
<td>This is not a &quot;hollow&quot; package:</td>
</tr>
<tr>
<td></td>
<td>- If installed on the global zone, the package content and installation</td>
</tr>
<tr>
<td></td>
<td>information are required on all non-global zones.</td>
</tr>
<tr>
<td></td>
<td>- The package delivers software that should be visible in all non-global</td>
</tr>
<tr>
<td></td>
<td>zones. An example is the package that delivers the truss command.</td>
</tr>
<tr>
<td></td>
<td>- Other than the restrictions for the current setting of the</td>
</tr>
<tr>
<td></td>
<td>SUNW_PKG_ALLZONES package parameter, no additional restrictions are</td>
</tr>
<tr>
<td></td>
<td>defined.</td>
</tr>
</tbody>
</table>
TABLE 24–3 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.  
  ■ 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 removed from the global zone, the package is recognized as having been completely installed. Appropriate directories and files are removed, and class action or other install scripts are run when the package is removed.  
  ■ When removed from a non-global zone, the package is recognized as not having been completely installed. No directories are removed, no files are removed, and no class action or other install scripts are run when the package is removed.  
  ■ 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. |
### TABLE 24–4  SUNW_PKG_THISZONE Package Parameter Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| false | ■ If pkgadd is run in a non-global zone, the package is installed in the current zone only.  
■ If pkgadd is run in the global zone, the package is installed in the global zone and also installed in all currently installed non-global zones. In addition, the package will be propagated to all future, newly installed non-global zones. |
| true | ■ The package is installed in the current zone only.  
■ If installed in the global zone, the package is not added to any currently existing or yet-to-be-created non-global zones. This is the same behavior that occurs when the -g option is specified to pkgadd. |

### 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" on page 304.

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.

### About Adding Patches in Zones

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.

**Note** – If a package is installed with `pkgadd -G` or has the `pkginfo` setting `SUNW_PKG_THISZONE=true`, the package *can only* be patched with `patchadd -G`.

### Applying Patches on a Solaris System With Zones Installed

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 `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.

**Removing Patches on a Solaris System With Zones Installed**

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**

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**

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)

**Solaris 10 1/06:** In this release, this chapter has been completely revised. This chapter documents current package and patch procedures on a system with installed non-global zones.

**Solaris 10 6/06:** A note was added to the procedure "How to Add a Package to the Global Zone Only" on page 306.

For a complete listing of new Solaris 10 features and a description of Solaris releases, see *Solaris 10 What's New*.

This chapter describes how to add and remove packages and patches on a system 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 with zones installed, see Chapter 24.

## Adding and Removing Packages and Patches on a Solaris System With Zones Installed (Task Map)

<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 306</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 309</td>
</tr>
</tbody>
</table>
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 Packages and Zones” on page 287. For more information about how to check these package parameter settings, see “Checking Package Parameter Settings on a System with Zones Installed” on page 313.

▼ 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 `pkgrm` 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 `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:
     ```bash
     my-zone# pkgadd -d /cdrom/cdrom0/directory package_name
     ```
   - If installing the package from a directory to which it has been copied, type:
     ```bash
     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 Packages and Zones" on page 287. For more information about how to check these package parameter settings, see "Checking Package Parameter Settings on a System with Zones Installed" on page 313.

▼ 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

**Note** – If you are patching a package that was added by using the `pkgadd` command with the `-G` option, the package must be patched by using the `patchadd` command with the `-G` option.

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.

2. While in the non-global zone, 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, \texttt{my-zone} in this procedure, execute the \texttt{patchrm} command followed by the patch ID.

\texttt{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 \texttt{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 \texttt{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 \texttt{pkgparam} followed by the package name and the name of the parameter.

\texttt{my-zone% pkgparam package_name SUNW_PKG_ALLZONES true}

\texttt{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 \texttt{pkgparam -d} with the path of the CD-ROM followed by the package name and the name of the parameter.

\texttt{my-zone% pkgparam -d /cdrom/cdrom0/directory package_name SUNW_PKG_ALLZONES true}

\texttt{my-zone% pkgparam -d /cdrom/cdrom0/directory package_name SUNW_PKG_HOLLOW false}
This chapter covers these general zone administration topics:

- “What’s New in This Chapter?” on page 315
- “Global Zone Visibility and Access” on page 316
- “Process ID Visibility in Zones” on page 316
- “System Observability in Zones” on page 317
- “Non-Global Zone Node Name” on page 317
- “File Systems and Non-Global Zones” on page 317
- “Networking in Non-Global Zones” on page 324
- “Device Use in Non-Global Zones” on page 326
- “Running Applications in Non-Global Zones” on page 328
- “Resource Controls Used in Non-Global Zones” on page 328
- “Fair Share Scheduler on a Solaris System With Zones Installed” on page 329
- “Extended Accounting on a Solaris System With Zones Installed” on page 330
- “Privileges in a Non-Global Zone” on page 330
- “Using IP Security Architecture in Zones” on page 333
- “Using Solaris Auditing in Zones” on page 334
- “Core Files in Zones” on page 335
- “Commands Used on a Solaris System With Zones Installed” on page 339

What’s New in This Chapter?

**Solaris 10 1/06:** A new section “Unmounting File Systems in Zones” on page 320 has been added.

**Solaris 10 1/06:** New sections on zone backup and restore procedures have been added. See “About Backing Up a Solaris System With Zones Installed” on page 335.

**Solaris 10 6/06:** A ZFS entry has been added to the table in “Mounting File Systems in Zones” on page 318.
For a complete listing of new Solaris 10 features and a description of Solaris releases, see *Solaris 10 What’s New*.

**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)
- ptree (see the proc(1) man page)
- prstat (see the prstat(1M) man page)

See Table 26–4 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 -o nosuid 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 listed in "Mounting File Systems in Zones" on page 318. Mounting capabilities are also described. For more information about the -o nosuid option, see "Accessing Network File Systems (Reference)" in System Administration Guide: Network Services.

### Mounting File Systems in Zones

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 233 and "Mounting File Systems in Running Non-Global Zones" on page 351.

<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>
</tbody>
</table>

**Note** – 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.

**Note** – 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>HSFS</td>
<td>Can be mounted using <code>zonecfg</code>, 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 <code>zonecfg</code>, 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 <code>zonecfg</code>, cannot be manually mounted from the global zone into a non-global zone. Can be mounted from within the zone.</td>
</tr>
<tr>
<td>NFS</td>
<td>Cannot be mounted using <code>zonecfg</code>. V2, V3, and V4, which are the versions currently supported in zones, can be mounted from within the zone.</td>
</tr>
<tr>
<td>PCFS</td>
<td>Can be mounted using <code>zonecfg</code>, 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 <code>zonecfg</code>, 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 <code>zonecfg</code>, 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 <code>zonecfg</code>, 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 <code>zonecfg</code>, 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>Can be mounted using <code>zonecfg</code>, can be manually mounted from the global zone into a non-global zone, can be mounted from within the zone. This file system might not be included in a future release.</td>
</tr>
<tr>
<td>ZFS</td>
<td>Can be mounted using the <code>zonecfg dataset</code> and <code>fs resource types</code>.</td>
</tr>
</tbody>
</table>

For more information, see “How to Configure the Zone” on page 234, “Mounting File Systems in Running Non-Global Zones” on page 351, 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, the global zone owns this mount and the zone administrator for the non-global 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, the zone administrator in the non-global zone 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 Autofs 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 mnttab 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/mnttab 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 265.

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, HSFS, PCFS, and other storage-based file systems
When using the zonecfg command to configure storage-based file systems that have an fsck binary, such as UFS, the zone administrator must specify a raw parameter. The parameter indicates the raw (character) device, such as /dev/rdsk/c0t0d0s7. zoneadm 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, zoneadm 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/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 zoneadm Daemon” on page 249 and the fsck(1M)
ZFS
You can add a ZFS dataset to a non-global zone by using the zonecfg 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 zonecfg 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 8, “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 root 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
Networking in 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.

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 `SOCK_RAW` socket interfaces are restricted to Internet Control Message Protocol (ICMP). ICMP is necessary for detecting and reporting network error conditions or using the `ping` command.

**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 interfaces configured by `zonecfg` will automatically be plumbed 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 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.
### IP Network Multipathing in 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 Non-Global Zones” on page 358. The zones configuration procedure is covered in “How to Configure the Zone” on page 234. For information on IPMP features, components, and usage, see Chapter 30, "Introducing IPMP (Overview)," in *System Administration Guide: IP Services*.

### 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.
The /dev file hierarchy, which is today part of the / (root) file system, consists of symbolic links, or logical paths, to the physical paths present in /devices. Applications reference the logical path to a device presented in /dev. The /dev file system is loopback-mounted into the zone using a read-only mount.

The /dev file hierarchy is managed by a system comprised of the components in the following list:

- **devfsadm** (see the `devfsadm(1M)` man page)
- **syseventd** (see the `syseventd(1M)` man page)
- **libdevinfo** device information library (see the `libdevinfo(3LIB)` man page)
- **devinfo** driver (see the `devinfo(7D)` man page)
- **Reconfiguration Coordination Manager (RCM)** (see “Reconfiguration Coordination Manager (RCM) Script Overview” in *System Administration Guide: Devices and File Systems*)

---

**Caution** – Subsystems that rely on /devices path names are not able to run in non-global zones until /dev path names are established.

---

**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 or /dev/ip.
- 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.

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 non-global zone /etc/project file, NIS map, or LDAP directory service. The settings for a given non-global 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 `rcpd` running, 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. See “Using the Fair Share Scheduler on a Solaris System With Zones Installed” on page 359 for example procedures.

**FSS Share Division in a Non-Global Zone**

FSS CPU shares for a zone are hierarchical. The shares for a given non-global zone 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.

For more information on `project.cpu-shares`, see “Available Resource Controls” on page 76.

**Share Balance Between Zones**

The global 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 within a 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.

The `limitpriv` property is available beginning with the Solaris 10 11/06 release.

### 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>dtrace_user</td>
<td>Optional</td>
<td>profile and syscall providers</td>
</tr>
<tr>
<td>gart_access</td>
<td>Optional</td>
<td>ioctl(2) access to agpgart_io(7I)</td>
</tr>
<tr>
<td>gart_map</td>
<td>Optional</td>
<td>mmap(2) access to agpgart_io(7I)</td>
</tr>
<tr>
<td>net_rawaccess</td>
<td>Optional</td>
<td>Raw PF_INET/PF_INET6 packet access</td>
</tr>
<tr>
<td>Privilege</td>
<td>Status</td>
<td>Notes</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>proc_clock_highres</td>
<td>Optional</td>
<td>Use of high resolution timers</td>
</tr>
<tr>
<td>proc_lock_memory</td>
<td>Optional</td>
<td>Locking memory; shmct1(2) and mlock(3C)</td>
</tr>
<tr>
<td>proc_priocntl</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_suserCompat</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>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>
<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>
</tbody>
</table>
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>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_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 Trusted 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.

**Note** – Trusted Solaris privileges are interpreted only if the system is configured with Trusted Extensions.

TABLE 26–2 Status of Trusted Solaris Privileges in Zones

<table>
<thead>
<tr>
<th>Trusted Solaris Privilege</th>
<th>Status</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>
### TABLE 26–2  Status of Trusted Solaris Privileges in Zones  (Continued)

<table>
<thead>
<tr>
<th>Trusted Solaris Privilege</th>
<th>Status</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<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>
<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 via NFS</td>
</tr>
</tbody>
</table>

To alter privileges in a non-global zone configuration, see “Configuring, Verifying, and Committing a Zone” on page 233.

To inspect privilege sets, see “Using the `ppriv` Utility” on page 344. 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.
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 running the `ipseckey` and `ipsecconf` commands from the global zone. Use the source address that corresponds to the non-global zone that you are configuring.

For more information, see the `ipsecconf(1M)` and `ipseckey(1M)` man pages.

### Using Solaris Auditing in Zones

Solaris auditing is described in Chapter 27, “Solaris Auditing (Overview),” in *System Administration Guide: Security Services*. For zones considerations associated with auditing, see the following sections:

- Chapter 28, “Planning for Solaris Auditing,” in *System Administration Guide: Security Services*
- "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 334, 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.

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 the table Commands Used to Administer Zones provide the primary administrative interface to the zones facility.

**TABLE 26–3  Commands Used to Administer 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>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 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.

**TABLE 26–4  Commands Modified for Use on a Solaris System With Zones Installed**

<table>
<thead>
<tr>
<th>Command Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipcrm(1)</td>
<td>Added -z zone option. This option is only useful when the command is executed in the global zone.</td>
</tr>
<tr>
<td>ipcs(1)</td>
<td>Added -z zone option. This option is only useful when the command is executed in the global zone.</td>
</tr>
<tr>
<td>pgrep(1)</td>
<td>Added -z zoneidlist 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 zone for use with the -l option to list all privileges available in the current zone. Also use the option -v after zone to obtain verbose output.</td>
</tr>
<tr>
<td>Command Reference</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>priocntl(1)</strong></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><strong>proc(1)</strong></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>
<tr>
<td><strong>ps(1)</strong></td>
<td>Added <code>zonename</code> and <code>zoneid</code> to list of recognized format names used with the <code>-o</code> option. Added <code>-z zoneidlist</code> 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 <code>-Z</code> to print the name of the zone associated with the process. The name is printed under an additional column header, <code>ZONE</code>.</td>
</tr>
<tr>
<td><strong>renice(1)</strong></td>
<td>Added <code>zoneid</code> to list of valid arguments used with the <code>-i</code> option.</td>
</tr>
<tr>
<td><strong>sar(1)</strong></td>
<td>If executed in a non-global zone in which the pools facility is enabled, the <code>-b</code>, <code>-c</code>, <code>-g</code>, <code>-m</code>, <code>-p</code>, <code>-u</code>, <code>-w</code>, and <code>-y</code> options display values only for processors that are in the processor set of the pool to which the zone is bound.</td>
</tr>
<tr>
<td><strong>auditconfig(1M)</strong></td>
<td>Added <code>zonename</code> token.</td>
</tr>
<tr>
<td><strong>auditreduce(1M)</strong></td>
<td>Added <code>-z zone-name</code> option. Added ability to get an audit log of a zone.</td>
</tr>
<tr>
<td><strong>coreadm(1M)</strong></td>
<td>Added variable <code>%z</code> to identify the zone in which process executed.</td>
</tr>
<tr>
<td><strong>df(1M)</strong></td>
<td>Added <code>-Z</code> option to display mounts in all visible zones.</td>
</tr>
<tr>
<td><strong>ifconfig(1M)</strong></td>
<td>Added zone option for global zone use (the default), and <code>-zone zonename</code> for non-global zone use.</td>
</tr>
<tr>
<td><strong>iostat(1M)</strong></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><strong>mpstat(1M)</strong></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><strong>ndd(1M)</strong></td>
<td>When used in the global zone, displays information for all zones.</td>
</tr>
<tr>
<td><strong>netstat(1M)</strong></td>
<td>Displays information for the current zone only.</td>
</tr>
<tr>
<td><strong>nfsstat(1M)</strong></td>
<td>Displays statistics for the current zone only.</td>
</tr>
<tr>
<td><strong>poolbind(1M)</strong></td>
<td>Added <code>zoneid</code> list. Also see “Resource Pools Used in Zones” on page 137 for information about using zones with resource pools.</td>
</tr>
</tbody>
</table>
TABLE 26–4 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>prstat(1M)</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.</td>
</tr>
<tr>
<td>psrinfo(1M)</td>
<td>If executed in a non-global zone, only information about the processors visible to the zone is displayed.</td>
</tr>
<tr>
<td>traceroute(1M)</td>
<td>Usage change. When specified from within a non-global zone, the <code>-F</code> option has no effect because the &quot;don't fragment&quot; bit is always set.</td>
</tr>
<tr>
<td>vmstat(1M)</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>faults</code>, and <code>cpu</code> report fields.</td>
</tr>
<tr>
<td>auditon(2)</td>
<td>Added <code>AUDIT_ZONEID</code> to generate a zone ID token with each audit record.</td>
</tr>
<tr>
<td>priocntl(2)</td>
<td>Added <code>P_ZONEID</code> id argument.</td>
</tr>
<tr>
<td>processor_info(2)</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>p_online(2)</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>pset_bind(2)</td>
<td>Added <code>P_ZONEID</code> as idtype. Added zone to possible choices for <code>P_MYID</code> specification. Added <code>P_ZONEID</code> to valid idtype list in EINVAL error description.</td>
</tr>
<tr>
<td>pset_info(2)</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>pset_list(2)</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>pset_setattr(2)</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>sysinfo(2)</td>
<td>Changed <code>PRIV_SYS_CONFIG</code> to <code>PRIV_SYS_ADMIN</code>.</td>
</tr>
<tr>
<td>umount(2)</td>
<td>ENOENT is returned if file pointed to by <code>file</code> is not an absolute path.</td>
</tr>
<tr>
<td>getloadavg(3C)</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>Command Reference</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>getpriority(3C)</td>
<td>Added zone IDs to target processes that can be specified. Added zone ID to EINVAL error description.</td>
</tr>
<tr>
<td>priv_str_to_set(3C)</td>
<td>Added &quot;zone&quot; string for the set of all privileges available within the caller’s zone.</td>
</tr>
<tr>
<td>pset_getloadavg(3C)</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>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 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 n_type: NT_ZONENAME. 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 zone name audit policy is set.</td>
</tr>
<tr>
<td>privileges(5)</td>
<td>Added PRIV_PROC_ZONE, 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 cr.</td>
</tr>
</tbody>
</table>
This chapter covers general administration tasks and provides usage examples.

- "What's New in This Chapter?" on page 343
- "Using the ppriv Utility" on page 344
- "Mounting File Systems in Running Non-Global Zones" on page 351
- "Using IP Network Multipathing on a Solaris System With Zones Installed" on page 358
- "Using the Fair Share Scheduler on a Solaris System With Zones Installed" on page 359
- "Using Rights Profiles in Zone Administration" on page 360
- "Backing Up a Solaris System With Installed Zones" on page 361
- "Restoring a Non-Global Zone" on page 364

What's New in This Chapter?

This section lists new product features and identifies documentation improvements in this guide.

For a complete listing of new Solaris 10 features and a description of Solaris releases, see Solaris 10 What's New.

What's New in This Chapter for Solaris 10 1/06?

A new procedure for media access has been added. See "How to Add Access to CD or DVD Media in a Non-Global Zone" on page 354.

New procedures for backing up and restoring files in zones have been added. See "Backing Up a Solaris System With Installed Zones" on page 361 and "Restoring a Non-Global Zone" on page 364.
What's New in This Chapter for Solaris 10 6/06?

New procedures have been added. See "How to Mount a File System From the Global Zone Into a Non-Global Zone" on page 354 and "How to Add a Writable Directory under /usr in a Non-Global Zone" on page 356.

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.

  ```
  global# ppriv -l zone
  contract_event
  contract_observer
  cpc_cpu
  dtrace_kernel
  dtrace_proc
  dtrace_user
  file_chown
  file_chown_self
  file_dac_execute
  file_dac_read
  file_dac_search
  file_dac_write
  file downgrade_sl
  file_link_any
  file_owner
  file_setid
  file_upgrade_sl
  graphics_access
  graphics_map
  ipc dac_read
  ipc dac_write
  ipc owner
  net bindmlp
  net_icmpaccess
  net_mac_aware
  ```
Using the ppriv Utility

net_privaddr
net_rawaccess
proc_audit
proc_chroot
proc_clock_highres
proc_exec
proc_fork
proc_info
proc_lock_memory
proc_owner
proc_priocntl
proc_session
proc_setid
proc_taskid
proc_zone
sys_acct
sys_admin
sys_audit
sys_config
sys_devices
sys_ipc_config
sys_linkdir
sys_mount
sys_net_config
sys_nfs
sys_res_config
sys_resource
sys_suser_compat
sys_time
sys_trans_label
win_colormap
win_config
win_dac_read
win_dac_write
win_devices
win_dga
win_downgrade_sl
win_fontpath
win_mac_read
win_mac_write
win_selection
win_upgrade_sl
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  
   file_chown_self  
   file_dac_execute  
   file_dac_read  
   file_dac_search  
   file_dac_write  
   file_link_any  
   file_owner  
   file_setid  
   ipc_dac_read  
   ipc_dac_write  
   ipc_owner  
   net_bindmlp  
   net_icmpaccess  
   net_mac_aware  
   net_privaddr  
   proc_audit  
   proc_chroot  
   proc_exec  
   proc_fork  
   proc_info  
   proc_owner  
   proc_session  
   proc_setid  
   proc_taskid  
   sys_acct  
   sys_admin  
   sys_audit  
   sys_mount  
   sys_nfs  
   sys_resource  
   ```
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 -l -v zone
```

You will see a display similar to this:

- **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.

- **file_chown_self**
  - Allows a process to give away its files; a process with this privilege will run as if `{POSIX.CHOWN_RESTRICTED}` is not in effect.

- **file_dac_execute**
  - Allows a process to execute an executable file whose permission bits or ACL do not allow the process execute permission.

- **file_dac_read**
  - Allows a process to read a file or directory whose permission bits or ACL do not allow the process read permission.

- **file_dac_search**
  - Allows a process to search a directory whose permission bits or ACL do not allow the process search permission.

- **file_dac_write**
  - Allows a process to write a file or directory whose permission bits or ACL do not allow the process write permission.
  - In order to write files owned by uid 0 in the absence of an
effective uid of 0. ALL privileges are required.

**file_link_any**
Allows a process to create hardlinks to files owned by a uid different from the process’ effective uid.

**file_owner**
Allows a process which is not the owner of a file or directory to perform the following operations that are normally permitted only for the file owner: modify that file’s access and modification times; remove or rename a file or directory whose parent directory has the ‘‘save text image after execution’’ (sticky) bit set; mount a ‘‘namefs’’ upon a file; modify permission bits or ACL except for the set-uid and set-gid bits.

**file_setid**
Allows a process to change the ownership of a file or write to a file without the set-user-ID and set-group-ID bits being cleared.
Allows a process to set the set-group-ID bit on a file or directory whose group is not the process’ effective group or one of the process’ supplemental groups.
Allows a process to set the set-user-ID bit on a file with different ownership in the presence of PRIV_FILE_OWNER.
Additional restrictions apply when creating or modifying a set-uid 0 file.

**ipc_dac_read**
Allows a process to read a System V IPC Message Queue, Semaphore Set, or Shared Memory Segment whose permission bits do not allow the process read permission.
Allows a process to read remote shared memory whose permission bits do not allow the process read permission.

**ipc_dac_write**
Allows a process to write a System V IPC Message Queue, Semaphore Set, or Shared Memory Segment whose permission bits do not allow the process write permission.
Allows a process to read remote shared memory whose permission bits do not allow the process write permission.
Additional restrictions apply if the owner of the object has uid 0 and the effective uid of the current process is not 0.

**ipc_owner**
Allows a process which is not the owner of a System V IPC Message Queue, Semaphore Set, or Shared Memory Segment to remove, change ownership of, or change permission bits of the Message Queue, Semaphore Set, or Shared Memory Segment.
Additional restrictions apply if the owner of the object has uid 0 and the effective uid of the current process is not 0.

**net_bindmlp**
Allow a process to bind to a port that is configured as a multi-level port(MLP) for the process’s zone. This privilege
applies to both shared address and zone-specific address MLPs. See tzonecfg(4) from the Trusted Extensions manual pages for information on configuring MLP ports.
This privilege is interpreted only if the system is configured with Trusted Extensions.

net_icmpaccess
   Allows a process to send and receive ICMP packets.

net_mac_aware
   Allows a process to set NET_MAC_AWARE process flag by using setpflags(2). This privilege also allows a process to set SO_MAC_EXEMPT socket option by using setsockopt(3SOCKET). The NET_MAC_AWARE process flag and the SO_MAC_EXEMPT socket option both allow a local process to communicate with an unlabeled peer if the local process' label dominates the peer's default label, or if the local process runs in the global zone.
   This privilege is interpreted only if the system is configured with Trusted Extensions.

net_privaddr
   Allows a process to bind to a privileged port number. The privilege port numbers are 1-1023 (the traditional UNIX privileged ports) as well as those ports marked as "udp/tcp_extra_priv_ports" with the exception of the ports reserved for use by NFS.

proc_audit
   Allows a process to generate audit records.
   Allows a process to get its own audit pre-selection information.

proc_chroot
   Allows a process to change its root directory.

proc_exec
   Allows a process to call execve().

proc_fork
   Allows a process to call fork1()/forkall()/vfork()

proc_info
   Allows a process to examine the status of processes other than those it can send signals to. Processes which cannot be examined cannot be seen in /proc and appear not to exist.

proc_owner
   Allows a process to send signals to other processes, inspect and modify process state to other processes regardless of ownership. When modifying another process, additional restrictions apply: the effective privilege set of the attaching process must be a superset of the target process' effective, permitted and inheritable sets; the limit set must be a superset of the target's limit set; if the target process has any uid set to 0 all privilege must be asserted unless the effective uid is 0.
   Allows a process to bind arbitrary processes to CPUs.
Using the ppriv Utility

**proc_session**
- Allows a process to send signals or trace processes outside its session.

**proc_setid**
- Allows a process to set its uids at will.
  - Assuming uid 0 requires all privileges to be asserted.

**proc_taskid**
- Allows a process to assign a new task ID to the calling process.

**sys_acct**
- Allows a process to enable and disable and manage accounting through acct(2), getacct(2), putacct(2) and wracct(2).

**sys_admin**
- Allows a process to perform system administration tasks such as setting node and domain name and specifying nscd and coreadm settings.

**sys_audit**
- Allows a process to start the (kernel) audit daemon.
  - Allows a process to view and set audit state (audit user ID, audit terminal ID, audit sessions ID, audit pre-selection mask).
  - Allows a process to turn off and on auditing.
  - Allows a process to configure the audit parameters (cache and queue sizes, event to class mappings, policy options).

**sys_mount**
- Allows filesystem specific administrative procedures, such as filesystem configuration ioctl's, quota calls and creation/deletion of snapshots.
- Allows a process to mount and unmount filesystems which would otherwise be restricted (i.e., most filesystems except namefs).
- A process performing a mount operation needs to have appropriate access to the device being mounted (read-write for "rw" mounts, read for "ro" mounts).
- A process performing any of the aforementioned filesystem operations needs to have read/write/owner access to the mount point.
- Only regular files and directories can serve as mount points for processes which do not have all zone privileges asserted.
- Unless a process has all zone privileges, the mount(2) system call will force the "nosuid" and "restrict" options, the latter only for autofs mountpoints.
- Regardless of privileges, a process running in a non-global zone may only control mounts performed from within said zone.
- Outside the global zone, the "nodevices" option is always forced.

**sys_nfs**
- Allows a process to perform Sun private NFS specific system calls.
- Allows a process to bind to ports reserved by NFS: ports 2049 (nfs) and port 4045 (lockd).

**sys_resource**
Allows a process to modify the resource limits specified by `setrlimit(2)` and `setrctl(2)` without restriction. Allows a process to exceed the per-user maximum number of processes. Allows a process to extend or create files on a filesystem that has less than minfree space in reserve.

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.

\section*{How to Import Raw and Block Devices by Using \texttt{zonecfg}}

This procedure uses the \texttt{lofi} file driver, which exports a file as a block device.

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{Change directories to `/usr/tmp`.
   \texttt{global# cd /usr/tmp}}

3. \textbf{Create a new UFS file system.}
   \texttt{global# mkfile 10m fsfile}

4. \textbf{Attach the file as a block device.}
   The first available slot, which is `/dev/lofi/1` if no other \texttt{lofi} devices have been created, is used.
   \texttt{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/rlofi/
zonecfg:my-zone:device> end
zonecfg:my-zone> add device
zonecfg:my-zone:device> set match=/dev/lofi/
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/lofi/1

You will see a display that is similar to this:

** /dev/lofi/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/lofi/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/lofi/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 zonepath /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.

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
   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:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
</tr>
<tr>
<td>1</td>
<td>my-zone</td>
<td>running</td>
<td>/export/home/my-zone</td>
</tr>
</tbody>
</table>

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
Install the product as described in the product installation guide.

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.

(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 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
```
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.

```sh
zonecfg:my-zone> set type=lofs
zonecfg:my-zone> 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 Extend IP Network Multipathing Functionality to Non-Global Zones

Use this procedure to configure IP Network Multipathing (IPMP) in the global zone and extend its 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 216 and “How to Configure the Zone” on page 234 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 “Network Interfaces” on page 325 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: IP Services.

3 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:

```bash
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, then 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.

Using the Fair Share Scheduler on a Solaris System With Zones Installed

▼ How to Set FSS Shares in the Global Zone

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.

2 Use the prctl utility to assign two shares to the global zone:

   # prctl -n zone.cpu-shares -v 2 -r -i zone global

3 (Optional) To verify the number of shares assigned to the global zone, type:

   # prctl -n zone.cpu-shares -i zone global

See Also

For more information on the prctl utility, see the prctl(1) man page.

▼ How to Balance CPU Usage Between the Global Zone and Non-Global Zones

This procedure can be used to equalize the CPU usage between the global zone and non-global zones. In this procedure, a configuration with two non-global zones is used. The global zone receives one-third of the machine’s CPU resource. The non-global zones each receive one-third of the machine’s CPU resource.
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 to assign one share to each non-global zone as shown in Step 13.**
   Because the global zone receives one share by default, also giving each non-global zone one share divides the machine equally.

### How to Change the zone.cpu-shares Value in a Zone Dynamically

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 prctl command to specify a new value for cpu-shares.**
   
   # prctl -i idtype -n zone.cpu-shares -r -v value
   
   *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:

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td></td>
<td>/export/my-zone</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.ufsdump`. You must create the directory `backup` under `/`. 
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.

Create the snapshot.

global# fssnap -o bs=/export /export/home

You will see a display similar to the following:

    dev/fssnap/0

Mount the snapshot.

global# mount -o ro /dev/fssnap/0 /mnt

Backup 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

Unmount the snapshot.

global# umount /mnt

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

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.
### 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`.
   ```bash
   global# zonecfg -z my-zone -f my-zone.config
   ```

3 Install the zone.
   ```bash
   global# zoneadm -z my-zone install
   ```

4 To prevent the system from displaying the `sysidtool` questions upon initial zone login, delete the file `zonepath/root/etc/.UNCONFIGURED`, for example:
   ```bash
   global# rm /export/home/my-zone/root/etc/.UNCONFIGURED
   ```

5 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 provides information about how to upgrade your Solaris™ 10 system to a later release if you are running Solaris Zones. Links to appropriate Solaris installation documents are provided.

Read “Zones With an $fs Resource Defined With a Type of $lofs Cannot Be Upgraded to the Solaris 10 11/06 Release” on page 369 before upgrading your system.

**Backing Up Your System Before Performing an Upgrade**

You should back up the global and non-global zones on your Solaris system before you perform the upgrade. See "About Backing Up a Solaris System With Zones Installed" on page 335 and "Backing Up a Solaris System With Installed Zones" on page 361 for information.

**Information on Upgrading a Solaris 10 System With Installed Zones**

You can use either the standard Solaris interactive installation program or the custom JumpStart installation program to upgrade your Solaris system with zones installed. Solaris Live Upgrade is not supported for this release. For information, see Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning and Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations.

- The overall planning information and requirements for all types of installations and upgrades are documented in Chapter 3, “Solaris Installation and Upgrade (Planning),” in Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning. Note that the media used for the installation must be either a DVD or a network installation image created from a DVD.
- The Solaris 10 release interface is documented in Solaris 10 Installation Guide: Basic Installations.
The special considerations and limitations for custom JumpStart installations are described in Chapter 11, “Custom JumpStart (Reference),” in Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations.

Information on performing an installation or upgrade over the network is provided in Solaris 10 Installation Guide: Network-Based Installations.
Troubleshooting Miscellaneous Solaris Zones Problems

This chapter is new for the Solaris 10 6/06 release.

For a complete listing of new Solaris 10 features and a description of Solaris releases, see Solaris 10 What’s New.

Zones With an fs Resource Defined With a Type of lofs Cannot Be Upgraded to the Solaris 10 11/06 Release

If all non-global zones that are configured with lofs fs resources are mounting directories that exist in the miniroot, the system can be upgraded from an earlier Solaris 10 release to the Solaris 10 11/06 release using standard upgrade. For example, a lofs mounted /opt directory presents no issues for upgrade.

However, if any of your non-global zones are configured with a non-standard lofs mount, such as a lofs mounted /usr/local directory, the following error message is displayed:

The zones upgrade failed and the system needs to be restored from backup. More details can be found in the file /var/sadm/install_data/upgrade_log on the upgrade root file system.

Although this error message states that the system must be restored from backup, the system is actually fine, and it can be upgraded successfully using the following workaround:

1. Reboot your system with the installed OS.
2. Reconfigure the zones, removing the fs resources defined with a type of lofs.
3. After removing these resources, upgrade the system to Solaris 10 11/06.
4. Following the upgrade, you can reconfigure your zones again to restore the additional fs resources that you removed.
Solaris 10 6/06 and Solaris 10 11/06: Do Not Place the Root File System of a Non-Global Zone on ZFS

The zonepath of a non-global zone should not reside on ZFS for this release. This action might result in patching problems and possibly prevent the system from being upgraded to a later Solaris 10 update release.

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.
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 of the zone.

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:

```bash
zonecfg:zone5> set limitpriv="basic"

global# zoneadm -z zone5 boot
  required privilege "sys_mount" is missing from the zone's privilege set
zoneadm: zone zone5 failed to verify
```

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 256:

```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.

### Resolving Problems With a zoneadm attach Operation

#### Patches and Packages Are Out of Sync

The target system must be running the same 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

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:
```
To migrate the zone successfully, 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 and Chapter 25.

### Operating System Releases Do Not Match

To migrate the zone successfully, install the same Solaris release that is running on the original host.

1. **Verify the Solaris release running on the original system.**
   ```bash
   host1# uname -a
   ```

2. **Install the same release on the new host.**
   Refer to the Solaris installation documentation on docs.sun.com.
Glossary

bless
In Perl, the keyword used to create an object.

blessed
In Perl, the term used to denote class membership.

cap
A limit that is placed on system resource usage.

capping
The process of placing a limit on system resource usage.

default pool
The pool created by the system when pools are enabled.
See also resource pool.

default processor set
The processor set created by the system when pools are enabled.
See also processor set.

disjoint
A type of set in which the members of the set do not overlap and are not duplicated.

dynamic configuration
Information about the disposition of resources within the resource pools framework for a given system at a point in time.

dynamic reconfiguration
On SPARC based systems, the ability to reconfigure hardware while the system is running. Also known as DR.

extended accounting
A flexible way to record resource consumption on a task basis or process basis in the Solaris Operating System.

fair share scheduler
A scheduling class, also known as FSS, that allows you to allocate CPU time that is based on shares. Shares define the portion of the system's CPU resources allocated to a project.

FSS
See fair share scheduler.

global administrator
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.
Glossary

**global scope**  
Actions that apply to resource control values for every resource control on the system.

**global zone**  
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](#).

**heap**  
Process-allocated scratch memory.

**local scope**  
Local actions taken on a process that attempts to exceed the control value.

**locked memory**  
Memory that cannot be paged.

**memory cap enforcement threshold**  
The percentage of physical memory utilization on the system that will trigger cap enforcement by the resource capping daemon.

**naming service database**  
In the Projects and Tasks (Overview) chapter of this document, a reference to both LDAP containers and NIS maps.

**non-global zone**  
A virtualized operating system environment created within a single instance of the Solaris Operating System. The Solaris Zones software partitioning technology is used to virtualize operating system services.

**non-global zone administrator**  
See zone administrator.

**page in**  
To read data from a file into physical memory one page at a time.

**page out**  
To relocate pages to an area outside of physical memory.

**pool**  
See [resource pool](#).

**pool daemon**  
The poold system daemon that is active when dynamic resource allocation is required.

**processor set**  
A disjoint grouping of CPUs. Each processor set can contain zero or more processors. A processor set is represented in the resource pools configuration as a resource element. Also referred to as a pset.

See also [disjoint](#).

**project**  
A network-wide administrative identifier for related work.

**resident set size**  
The size of the resident set. The resident set is the set of pages that are resident in physical memory.
| **resource** | An aspect of the computing system that can be manipulated with the intent to change application behavior. |
| **resource capping daemon** | A daemon that regulates the consumption of physical memory by processes running in projects that have resource caps defined. |
| **resource consumer** | Fundamentally, a Solaris process. Process model entities such as the project and the task provide ways of discussing resource consumption in terms of aggregated resource consumption. |
| **resource control** | A per-process, per-task, or per-project limit on the consumption of a resource. |
| **resource management** | A functionality that enables you to control how applications use available system resources. |
| **resource partition** | An exclusive subset of a resource. All of the partitions of a resource sum to represent the total amount of the resource available in a single executing Solaris instance. |
| **resource pool** | A configuration mechanism that is used to partition machine resources. A resource pool represents an association between groups of resources that can be partitioned. |
| **resource set** | A process-bindable resource. Most often used to refer to the objects constructed by a kernel subsystem offering some form of partitioning. Examples of resource sets include scheduling classes and processor sets. |
| **RSS** | See resident set size. |
| **scanner** | A kernel thread that identifies infrequently used pages and relocates the pages to an area outside of physical memory. |
| **Solaris Container** | A complete runtime environment for applications. Resource management and Solaris Zones software partitioning technology are both parts of the container. |
| **Solaris Zones** | A software partitioning technology used to virtualize operating system services and provide an isolated, secure environment in which to run applications. |
| **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. |
Glossary

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, 67
activating extended accounting, 66-68
administering resource pools, 156
attribute, project.pool, 141

B
binding to resource pool, 176
booting a zone, 256

C
changing resource controls temporarily, 85
cloning a zone, 251, 262
commands
   extended accounting, 61
   fair share scheduler, 109
   projects and tasks, 41
   resource controls, 86
   zones, 339
configurable privileges, zone, 215
configuration, rcapd, 119
configuring resource controls, 76
configuring zones, tasks, 227
CPU share configuration, 104
creating resource pools, 142

D
default processor set, 137
default project, 35
default resource pool, 137
deleting a zone, 263
disabling dynamic resource pools, 159
disabling resource capping, 130
disabling resource pools, 159
displaying extended accounting status, 67
DRP, 137
dynamic pools configuration, 139
dynamic resource pools
   disabling, 159
   enabling, 159

E
enabling dynamic resource pools, 159
enabling resource capping, 129
enabling resource pools, 159
entry format, /etc/project file, 37
/etc/project
   entry format, 37
   file, 36
/etc/user_attr file, 35
exact file, 58
extended accounting
   activating, 66-68
   chargeback, 58
commands, 61
file format, 58
extended accounting (Continued)
  overview, 58
  status, displaying, 67

F
fair share scheduler
  and processor sets, 105
  project.cpu-shares, 100
  share definition, 100
fair share scheduler (FSS), 100
FSS
  See fair share scheduler (FSS)
  configuration, 112

G
global administrator, 204, 206
  global zone, 204

H
  halting a zone, 250, 259
  troubleshooting, 250

I
  implementing resource pools, 141
  installing a zone, 255
  installing zones, 254
  interactive packages, 288
  interprocess communication, See resource controls
  IPC, 75

L
  lexacct, 58
  listing zones, 255
  login, remote zone, 267

M
  memory cap enforcement threshold, 120
  migrating a zone, 280
  moving a zone, 279-280

N
  non-global zone, 204

P
  package operations, 288
  packages, interactive, 288
  PAM (pluggable authentication module), identity
    management, 36
  patches generated for packages, 288
  Perl interface, 61
  pluggable authentication module, See PAM
  pool
    asynchronous control violation, 153
    configurable features, 149
    constraints, 144
    control scope, 152
    cpu-pinned property, 145
    description, 143
    dynamic resource allocation, 137
    logging information, 149
    objectives, 145
    synchronous control violation, 153
  pools, 136
  poolstat
    description, 154
    output format, 154
    usage examples, 178
  populating a zone, 248
  privilege levels, 80
  project
    active state, 101
    definition, 34
    idle state, 101
    with zero shares, 100
  project0, 104
  project.cpu-shares, 104
project database, 36
project.pool attribute, 141
project.system, See project 0
putacct, 59

R
rcap.max-rss, 119
rcapadm, 119
rcapd, 118
sample intervals, 123
scan intervals, 123
rcapd configuration, 119
rcapstat, 123
rctls, 74
See resource controls
ready a zone, 256
rebooting a zone, 260
remote zone login, 267
removing resource pools, 176
resource cap, 118
resource capping
disabling, 130
enabling, 129
resource capping daemon, 118
resource controls
changing temporarily, 85
configuring, 76
definition, 74
global actions, 81
inf value, 84
interprocess communication, 75
list of, 76
local actions, 81, 376
overview, 74
temporarily updating, 85
threshold values, 81, 376
zone-wide, 224
resource limits, 74
resource management
coststraints, 27
definition, 25
partitioning, 28
scheduling, 27
resource pools, 136
activating configuration, 175
administering, 156
binding to, 176
configuration elements, 140
creating, 142
disabling, 159
dynamic reconfiguration, 142
enabling, 159
/etc/pooladm.conf, 139
implementing, 141
properties, 140
removing, 176
removing configuration, 175
static pools configuration, 139
rlimits, See resource limits

S
server consolidation, 29
setting resource pool attributes, 176
Solaris Management Console
definition, 190
performance monitoring, 191
setting resource controls, 197
sparse root zone, 202
SUNW_PKG_ALLZONES package parameter, 297
SUNW_PKG_HOLLOW package parameter, 299
SUNW_PKG_THISZONE package parameter, 300

T
tasks, resource management, 40
temporarily updating resource controls, 85
threshold values, 80

U
uninstalling a zone, 261
V
/var/adm/exacct directory, 60
verifying a zone, 254

W
whole root zone, 202

Z
zone
adding packages, 289
adding patches, 301
boot procedure, 256
boot single-user, 258
clone, 251, 262
commands used in, 339
configurable privileges, 215
configuration, 216
creating, 206
definition, 201
delete, 263
disk space, 228
features, 210
halt procedure, 259
halt, 250
installation, 255
interactive mode, 267
list, 255
migrate, 280
move, 279-280
network address, 231
non-interactive mode, 268
package and patch overview, 286
package rules, 288
PatchPro support, 304
populating, 248
privileges, 330
ready state, 256
reboot, 251
reboot procedure, 260
removing packages, 292
removing patches, 303
zone (Continued)
resource controls, 224
resource type properties, 223
resource types, 219
scope, 286
states, 207
uninstall procedure, 261
verify, 254
zone administrator, 206
zone configuration
overview, 214
script, 238
tasks, 227
zone console login, console login mode, 266
zone.cpu-shares, 224
zone resource control, 219
zone host name, 230
zone ID, 204
zone installation
overview, 247
tasks, 254
zone login
failsafe mode, 267
overview, 265
remote, 267
zone.max-lwps, 224
zone resource control, 219
zone name, 204
zone node name, 317
zone resource controls, 219
zone root file system models, 202
zone size, restricting, 230
zone-wide resource controls, 224
zoneadm, 249
zonecfg
entities, 219
modes, 217
operations, 214
procedure, 233
scope, 217
scope, global, 217
scope, resource specific, 217
subcommands, 217
zones, characteristics by type, 205
zones commands, 339
zsched, 250