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

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

Si ce logiciel, ou la documentation qui l’accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l’utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s’applique:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

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

### About the Oracle ZFS Storage Appliance
- Oracle ZFS Storage Appliance Key Features .......................................................... 17
- Supported Protocols ............................................................................................ 18
- Oracle ZFS Storage Appliance Data Services ..................................................... 18
- Data Availability .................................................................................................. 19
- Oracle ZFS Storage Appliance Configuration ...................................................... 20
- Browser User Interface (BUI) ........................................................................... 20
- Network Icons .................................................................................................... 28
- Dashboard Icons ............................................................................................... 28
- Analytics Toolbar Icons ..................................................................................... 29
- Identity Mapping Icons ....................................................................................... 30
- Supported Browsers ........................................................................................... 31

### Command Line Interface (CLI)
- CLI Contexts ...................................................................................................... 32
- CLI Properties .................................................................................................... 38

### Working with CLI Scripting
- Using Batch Commands ..................................................................................... 41
- Understanding the CLI Scripting Commands ...................................................... 41
- ▼ Accessing the CLI Script Environment ................................................................. 42
- Understanding the Built-in CLI Functions .......................................................... 42
- ▼ Using the Run Function ................................................................................... 43
- ▼ Using the Get Function ................................................................................... 43
- ▼ Using the List Function ................................................................................... 44
- ▼ Using the Children Function ........................................................................... 45
- ▼ Using the Choices Function ............................................................................ 46
- Using the Functions for Generating Output ......................................................... 47
- Understanding CLI Scripting Errors .................................................................... 47

### Configuring the Oracle ZFS Storage Appliance
- Initial Appliance Configuration ........................................................................... 49
- ▼ Initial Configuration Using the BUI ................................................................. 50
Discovering FC Ports using the BUI ........................................................ 107
Creating FC Initiator Groups using the BUI .............................................. 108
Associating a LUN with an FC Initiator Group using the BUI ................. 109
▼ Changing FC Port Modes using the CLI .............................................. 110
▼ Discovering FC Ports using the CLI ................................................... 110
▼ Creating FC Initiator Groups using the CLI ......................................... 111
▼ Associating a LUN with an FC Initiator Group using the CLI ............... 112
▼ Scripting Aliases for Initiators and Initiator Groups using the CLI ...... 112
Configuring SAN iSCSI ....................................................................... 114
Configuring SAN iSCSI Initiators .......................................................... 115
▼ Creating an Analytics Worksheet using the BUI .................................. 116
▼ Configuring SAN iSER Targets ......................................................... 116
▼ Adding an iSCSI Target with an Auto-generated IQN using the CLI ...... 119
▼ Adding an iSCSI Target with a Specific IQN and RADIUS Authentication using the CLI ................................................................. 120
▼ Adding an iSCSI Initiator with CHAP Authentication using the CLI ...... 120
▼ Adding an iSCSI Target Group using the CLI ..................................... 121
▼ Adding an iSCSI Initiator Group using the CLI ................................... 121
Configuring SAN SRP ....................................................................... 121
▼ Configuring SRP Target using the BUI .............................................. 122
▼ Configuring SRP Targets Using the CLI .......................................... 123
Configuring Users ............................................................................... 124
User Authorizations .......................................................................... 125
Managing User Properties .................................................................. 126
▼ Adding an Administrator using the BUI .......................................... 127
▼ Adding a Role using the BUI ............................................................ 128
▼ Adding Authorizations to a Role using the BUI ................................. 128
▼ Deleting Authorizations from a Role using the BUI ......................... 128
▼ Adding a User Who can Only View the Dashboard using the BUI ...... 128
▼ Adding a Role using the CLI ............................................................ 129
▼ Adding an Administrator using the CLI .......................................... 129
▼ Adding Authorizations to a Role using the CLI ................................. 130
▼ Deleting Authorizations from a Role using the CLI ......................... 132
Setting Oracle ZFS Storage Appliance Preferences .................................. 132
Preference Properties ....................................................................... 132
▼ Setting Preferences using the CLI .................................................... 133
▼ Setting SSH Public Keys using the CLI ........................................... 134
Configuring Alerts ............................................................................. 134
Alert Categories ............................................................................... 135
Threshold Alerts ............................................................................. 135
Contents

▼ Adding a Threshold Alert using the BUI ............................................. 136
▼ Adding an Alert Action using the BUI ................................................ 137
▼ Adding a Threshold Alert using the CLI .............................................. 137
▼ Adding an Alert Action using the CLI ................................................ 138
Sending Email Alerts ............................................................................ 139
Sending an SNMP Trap ......................................................................... 139
Sending Syslog Messages ...................................................................... 140
Resuming/Suspending Datasets ............................................................... 140
Resuming/Suspending Worksheets ........................................................... 140
Executing a Workflow ........................................................................... 140

Configuring Clustering .................................................................................. 141
Understanding Clustering ....................................................................... 141
▼ Configuring Clustering using the BUI ................................................. 162
▼ Unconfiguring Clustering using the BUI .............................................. 164
▼ Shutting Down a Clustered Configuration using the CLI ......................... 165
▼ Shutdown the Stand-by Head using the CLI ......................................... 166
▼ Unconfiguring Clustering using the CLI .............................................. 166
Cabling ZS3-2 Clusters ......................................................................... 168
Cabling ZS4-4, ZS3-4, and 7x20 Clusters ................................................. 169
Cabling Storage Shelves for Clustering .................................................... 169

Working with Services ............................................................................... 173
Working with Oracle ZFS Storage Appliance Services ........................................ 174
Data Services ...................................................................................... 175
Required Service Ports .......................................................................... 177
Services BUI Page Icons and Buttons ...................................................... 177
Configuring Services using the CLI ......................................................... 178
▼ Viewing a Specific Service Screen using the BUI .................................. 179
▼ Viewing a Specific Service Screen ..................................................... 180
▼ Enabling a Service using the BUI ...................................................... 180
▼ Disabling a Service using the BUI ...................................................... 180
▼ Defining Properties using the BUI ..................................................... 180
▼ Viewing Service Logs using the BUI .................................................. 180
▼ Selecting a Service using the CLI ....................................................... 181
▼ Viewing a Service's State using the CLI .............................................. 181
▼ Enabling a Service using the CLI ...................................................... 181
▼ Disabling a Service using the CLI ...................................................... 182
▼ Setting Properties using the CLI ....................................................... 182
▼ Viewing Service Help using the CLI .................................................. 182
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working with NFS</td>
<td>183</td>
</tr>
<tr>
<td>NFS Properties</td>
<td>183</td>
</tr>
<tr>
<td>Configuring Kerberos Realms for NFS</td>
<td>185</td>
</tr>
<tr>
<td>NFS Logs</td>
<td>186</td>
</tr>
<tr>
<td>NFS Analytics</td>
<td>186</td>
</tr>
<tr>
<td>NFS Properties</td>
<td>187</td>
</tr>
<tr>
<td>▼ Sharing a Filesystem over NFS</td>
<td>187</td>
</tr>
<tr>
<td>Working with iSCSI</td>
<td>187</td>
</tr>
<tr>
<td>Working with SMB</td>
<td>189</td>
</tr>
<tr>
<td>SMB Properties</td>
<td>190</td>
</tr>
<tr>
<td>SMB Share Properties</td>
<td>191</td>
</tr>
<tr>
<td>NFS/SMB Interoperability</td>
<td>192</td>
</tr>
<tr>
<td>SMB DFS Namespaces</td>
<td>192</td>
</tr>
<tr>
<td>SMB Microsoft Stand-alone DFS Namespace Management Tools Support Matrix</td>
<td>193</td>
</tr>
<tr>
<td>▼ Adding DFS Namespaces to a Local SMB Group</td>
<td>194</td>
</tr>
<tr>
<td>SMB Autohome</td>
<td>195</td>
</tr>
<tr>
<td>▼ Adding SMB Autohome Rules</td>
<td>195</td>
</tr>
<tr>
<td>SMB Local Groups</td>
<td>196</td>
</tr>
<tr>
<td>▼ Adding a User to an SMB Local Group</td>
<td>197</td>
</tr>
<tr>
<td>SMB MMC Integration</td>
<td>197</td>
</tr>
<tr>
<td>SMB Share Management</td>
<td>198</td>
</tr>
<tr>
<td>SMB Users, Groups, and Connections</td>
<td>199</td>
</tr>
<tr>
<td>Listing SMB Services</td>
<td>200</td>
</tr>
<tr>
<td>▼ Configuring SMB using the BUI</td>
<td>202</td>
</tr>
<tr>
<td>▼ Configuring SMB Active Directory using the BUI</td>
<td>203</td>
</tr>
<tr>
<td>▼ Configuring SMB Project and Share using the BUI</td>
<td>204</td>
</tr>
<tr>
<td>▼ Configuring SMB Data Service using the BUI</td>
<td>204</td>
</tr>
<tr>
<td>Working with FTP</td>
<td>205</td>
</tr>
<tr>
<td>FTP Properties</td>
<td>205</td>
</tr>
<tr>
<td>FTP Logs</td>
<td>206</td>
</tr>
<tr>
<td>Working with HTTP</td>
<td>206</td>
</tr>
<tr>
<td>HTTP Properties</td>
<td>207</td>
</tr>
<tr>
<td>HTTP Authentication and Access Control</td>
<td>207</td>
</tr>
<tr>
<td>HTTP Logs</td>
<td>208</td>
</tr>
<tr>
<td>▼ Allowing HTTP Access to a Share using the BUI</td>
<td>208</td>
</tr>
<tr>
<td>Working with NDMP</td>
<td>208</td>
</tr>
<tr>
<td>NDMP Local vs. Remote Configurations</td>
<td>209</td>
</tr>
<tr>
<td>NDMP Backup Formats and Types</td>
<td>209</td>
</tr>
<tr>
<td>NDMP Back up with &quot;dump&quot; and &quot;tar&quot;</td>
<td>210</td>
</tr>
</tbody>
</table>
Identity Mapping Rules ................................................................. 234
Mapping Rule Directional Symbols ............................................... 234
Identity Mapping Best Practices ................................................... 235
Identity Mapping Concepts .......................................................... 235
▼ Configuring Identity Mapping using the BUI ................................. 237
▼ Viewing Mappings using the BUI ................................................ 238
▼ Flushing Mappings using the BUI ............................................... 238

Working with DNS ..................................................................... 239
DNS Properties ........................................................................... 239
Configuring DNS ....................................................................... 240
DNS Logs .................................................................................. 240
Active Directory and DNS ........................................................... 240
Non-DNS Resolution .................................................................. 240
DNS-Less Operation ................................................................... 241

Working with Dynamic Routing .................................................... 241
Working with IPMP ..................................................................... 241
Working with NTP ....................................................................... 242
NTP Properties ........................................................................... 243
▼ Setting Clock Synchronization using the BUI .............................. 244
▼ Configuring NTP using the CLI ............................................... 244

Working with Phone Home .......................................................... 245
Phone Home Properties ............................................................... 246
▼ Registering the Appliance using the BUI ................................. 247
▼ Registering the Appliance using the CLI ................................. 247
▼ Changing Account Information using the BUI .......................... 248

Working with the RESTful API ...................................................... 248
Working with Service Tags .......................................................... 248
Working with SMTP ..................................................................... 249
Working with SNMP .................................................................... 250
SNMP Properties ....................................................................... 250
SNMP MIBs ............................................................................... 251
Sun FM MIB .............................................................................. 251
Sun AK MIB ............................................................................ 252
▼ Configuring SNMP to Serve Appliance Status using the BUI ... 252
▼ Configuring SNMP to Send Traps using the BUI .................... 252

Working with Syslog ................................................................. 253
Syslog Properties ....................................................................... 254
Classic Syslog: RFC 3164 .......................................................... 254
Updated Syslog: RFC 5424 ......................................................... 254
SYSLOG Message Format ......................................................... 255
SYSLOG Alert Message Format ............................................................. 255
Example Configuring a Solaris Receiver using the CLI .......................... 257
Example Configuring a Linux Receiver using the CLI ............................ 257
Working with System Identity ................................................................. 258
Working with SSH .................................................................................. 259
SSH Properties ...................................................................................... 259
SSH Logs ............................................................................................... 259
▼ Disabling root SSH Access using the CLI ............................................ 260
Maintaining the Oracle ZFS Storage Appliance ........................................ 261
Working with Maintenance Workflows .................................................... 261
Understanding Workflows ...................................................................... 262
Understanding Workflow Parameters ...................................................... 263
Constrained Workflow Parameters .......................................................... 264
Optional Workflow Parameters ............................................................... 265
Workflow Error Handling ........................................................................ 266
Workflow Input Validation ....................................................................... 267
Workflow Execution Auditing and Reporting ............................................ 268
Understanding Workflow Versioning ....................................................... 270
Using Workflows for Alert Actions .......................................................... 271
Using Scheduled Workflows ................................................................. 273
Using a Scheduled Workflow .................................................................. 273
Coding Workflow Schedules ................................................................... 274
Creating a Worksheet Based on a Specified Drive Type .......................... 276
Uploading Workflows using the BUI ....................................................... 278
▼ Downloading Workflows using the CLI ................................................. 279
▼ Listing Workflows using the CLI ......................................................... 279
▼ Executing Workflows using the CLI .................................................... 280
Working with Shares .............................................................................. 283
Working with Storage Pools ................................................................. 284
Share Properties ..................................................................................... 285
Working with Snapshots ......................................................................... 286
Working with Clones ............................................................................. 286
Space Management for Shares ............................................................... 287
Shares Terminology ............................................................................... 287
Understanding Snapshots ....................................................................... 288
File System and Project Settings ............................................................. 289
▼ Viewing Current Shares Usage in the BUI .......................................... 290
Contents

- Viewing Current Shares Usage in the CLI ........................................... 291
- Setting User or Group Quotas .......................................................... 291
- Setting User or Group Quotas using the BUI ..................................... 292
- Setting User or Group Quotas using the CLI ..................................... 293
- Working with Identity Management .................................................. 294
- Working with Filesystem Namespace .................................................. 294
- Working with the Shares > Shares BUI Page ...................................... 295
- Understanding the Shares List ............................................................ 296
- Editing a Share .................................................................................. 297
- Share Usage Statistics ....................................................................... 297
- Share Static Properties ....................................................................... 298
- Using the Shares Project Panel ........................................................... 300
- Creating a Share (BUI) ...................................................................... 301
- Create Filesystem Parameters ............................................................. 301
- Create LUN Parameters ..................................................................... 303
- Selecting Shares (CLI) ....................................................................... 305
- Share Operations (CLI) ....................................................................... 306
- Shares CLI Properties ......................................................................... 307
- Working with the Shares > Shares > General BUI Page ...................... 309
- Shares > Shares > General BUI Page Settings ..................................... 310
- Understanding the Shares > Shares > Protocols BUI Page ................. 314
- Share Protocols - NFS ....................................................................... 315
- Configuring Share NFS Protocols using the CLI ............................... 316
- Configuring Share NFS Protocol Security Modes ............................. 317
- Configuring Share NFS Protocol Character Set Encodings ............... 318
- Shares Protocols - SMB ..................................................................... 319
- Shares Protocols - iSCSI .................................................................... 320
- Shares Protocols - HTTP ................................................................... 321
- Shares Protocols - FTP ...................................................................... 321
- Shares Protocols - SFTP .................................................................... 321
- Working with the Shares > Shares > Access BUI Page ...................... 321
- Shares - Root Directory Access ......................................................... 321
- Shares - Selecting Permissions ......................................................... 322
- Shares - ACL Behavior on Mode Change ......................................... 323
- Shares - ACL Inheritance Behavior ................................................... 324
- Shares - Root Directory ACL ............................................................... 325
- Shares - Snapshots ............................................................................ 328
- Listing Snapshots using the BUI ......................................................... 329
- Create a Project Level Snapshot using the BUI ................................. 330
- Create a Share/LUN Level Snapshot using the BUI ........................... 330
Contents

- Renaming a Snapshot using the BUI .......................................................... 330
- Destroying a Snapshot using the BUI ..................................................... 331
- Rolling back to a Snapshot using the BUI ............................................ 331
- Cloning a Snapshot using the BUI .......................................................... 331
- Scheduling Snapshots using the BUI ....................................................... 332
- Manual Snapshots using the CLI ......................................................... 333
- Listing Snapshots using the CLI ........................................................... 333
- Taking Manual Snapshots using the CLI ................................................ 333
- Renaming a Snapshot using the CLI ...................................................... 334
- Destroying a Snapshot using the CLI .................................................... 334
- Rolling back to a Snapshot using the CLI ............................................. 334
- Cloning a Snapshot using the CLI ....................................................... 335
- Listing Dependent Clones using the CLI ............................................. 336
- Scheduled Snapshots using the CLI ..................................................... 336
- Setting a Scheduled Snapshot Label using the CLI ............................... 337

Working with Replication ........................................................................... 337

Understanding Replication ..................................................................... 339
Understanding Project Replication Targets ........................................... 340
Understanding Project Replication Actions and Packages .................... 341
Understanding Project Replication Storage Pools ................................. 342
Understanding Project vs. Share Replication ......................................... 343
Understanding Replication Configuration Details ................................... 343
Replication Authorizations .................................................................... 344
Replication Alerts and Audit Events ...................................................... 345
Understanding Replication and Clustering ............................................. 345
Understanding Replication Snapshots and Data Consistency ................. 346
Managing Replication Snapshots ......................................................... 346
Understanding iSCSI Configurations and Replication ............................ 348
Replicating Clones ............................................................................... 348
Using Replication Analytics ................................................................. 348
Understanding Replication Failures ....................................................... 349
Understanding Appliance Upgrades and Replication ............................ 351
Creating and Editing Replication Actions ................................................ 352
- Creating and Editing Replication Targets using the BUI ....................... 354
- Creating and Editing Replication Targets using the CLI ...................... 355
- Creating and Editing Replication Actions using the BUI ....................... 356
- Creating and Editing Replication Actions using the CLI ...................... 356
Managing Replication Packages ............................................................ 358
Managing Replication Packages using the BUI ...................................... 359
Managing Replication Packages using the CLI ...................................... 360
Canceling Replication Updates ............................................................... 361
Disabling a Replication Package ............................................................ 362
Cloning a Replication Package or Share .................................................. 362
Exporting Replicated Filesystems .......................................................... 363
Severing Replication ............................................................................... 364
Reversing the Direction of Replication ..................................................... 365
Destroying a Replication Package ........................................................... 366
▼ Reversing Replication using the BUI .................................................... 366
▼ Reversing Replication for Disaster Recovery using the BUI .................... 368
▼ Reversing Replication to Resume Replication from Production System using the BUI ...................................................................................... 369
▼ Forcing Replication to use a Static Route using the BUI ....................... 370
▼ Cloning a Received Replication Project using the CLI ............................ 373
Working with Shadow Migration ............................................................. 374
Understanding Shadow Migration ........................................................... 374
Creating a Shadow Filesystem ............................................................... 377
Managing Background Migration ............................................................ 377
Handling Migration Errors ..................................................................... 378
Monitoring Migration Progress ............................................................... 378
Canceling Migration ............................................................................... 379
Replicating Shadow File Systems ............................................................ 379
Migrating Local File Systems ................................................................. 380
Using Shadow Migration Analytics ........................................................ 380
▼ Testing Potential Shadow Migration using the CLI ................................ 381
▼ Migrating Data from an Active NFS Server using the CLI ...................... 381
Working with Projects ............................................................................ 382
Managing Projects using the BUI ............................................................ 382
Project Usage Statistics ......................................................................... 383
▼ Creating Projects using the BUI ........................................................... 384
▼ Navigating Projects using the CLI ......................................................... 384
▼ Managing Projects using the CLI ........................................................ 385
▼ Selecting a Cluster Pool using the CLI ................................................ 386
Project CLI Properties ........................................................................... 387
Shares Project General BUI Page ............................................................ 388
Project Access ...................................................................................... 389
Project Snapshots ................................................................................. 390
Working with Schemas .......................................................................... 391
Configuring Schemas using the BUI ......................................................... 391
## Contents

- Configuring a Schema using the BUI .................................................. 392
- Configuring Schemas using the CLI ................................................... 392

### Working with Data Encryption ......................................................... 394
- Data Encryption Workflow .............................................................. 395
- Configuring LOCAL Keystore Encryption (BUI) ................................ 395
- Configuring LOCAL Keystore Encryption (CLI) ............................... 397
- Configuring OKM Keystore Encryption (BUI) ................................. 398
- Configuring OKM Keystore Encryption (CLI) ................................. 399
- Creating an Encrypted Share (CLI) ............................................... 400
- Changing a Project Encryption Key (BUI) ....................................... 401
- Changing a Project Encryption Key (CLI) ....................................... 402
- Changing a Share Encryption Key (BUI) ........................................ 403
- Changing a Share Encryption Key (CLI) ........................................ 404
- Deleting an Encryption Key (BUI) .................................................. 405
- Deleting an Encryption Key (CLI) .................................................. 406
- Restoring a LOCAL Key (CLI) ......................................................... 407

### Encryption Properties ..................................................................... 409
- Managing Encryption Keys ............................................................. 410
- Performance Impact of Encryption .................................................. 412
- Encryption Key Life Cycle ............................................................... 413
- Backing up and Restoring Encrypted Data ........................................ 413
- Replicating an Encrypted Share ...................................................... 413

### Integrating Applications with the Oracle ZFS Storage Appliance .......... 415

#### Configuring the Oracle ZFS Storage Appliance for Oracle Database Clients 416
- Oracle Exadata Database Machine Backup ....................................... 416
- Configuring the Appliance for Exadata .......................................... 417
- Configuring Exadata for Appliance ................................................ 420

#### Oracle SuperCluster Backup .......................................................... 424
- Configuring the Appliance for SuperCluster Backup ........................ 424
- Configuring Oracle SPARC SuperCluster for Appliance Backup ......... 442

#### Oracle Intelligent Storage Protocol .............................................. 445
- Database Record Size ................................................................. 445
- Synchronous Write Bias Hint ......................................................... 445
- Analytics Breakdown by Database Name ....................................... 446
- OISP-Capable Protocols and Clients .............................................. 446

#### Appliance Network File System Plug-in for Oracle Solaris Cluster ...... 446
- Appliance Plug-in for Oracle Solaris Cluster Geographic Edition ........ 446
- Appliance Plug-in for Oracle Enterprise Manager System Monitoring ... 447
Configuring for Oracle Enterprise Manager Monitoring ......................... 447
Unconfiguring Oracle Enterprise Manager Monitoring .......................... 448
Appliance Plug-in for Oracle Virtual Machine Storage Connect ............... 448
Appliance Plug-in Provider for Volume Shadow Copy Service Software ........ 449
FC Support with Symantec Dynamic Multi-Pathing and Storage Foundation .. 449
Appliance Replication Adapter for VMware Site Recovery Manager .............. 450
About the Oracle ZFS Storage Appliance

The Oracle ZFS Storage Appliance (appliance) family of products provides efficient file and block data services to clients over a network, and a rich set of data services that can be applied to the data stored on the system.

For information about configuring and working with the Oracle ZFS Storage Appliance, see the following sections:
- “Oracle ZFS Storage Appliance Key Features” on page 17
- “Supported Protocols” on page 18
- “Oracle ZFS Storage Appliance Data Services” on page 18
- “Data Availability” on page 19
- “Oracle ZFS Storage Appliance Configuration” on page 20
- “Browser User Interface (BUI)” on page 20
- “Network Icons” on page 28
- “Dashboard Icons” on page 28
- “Analytics Icons” on page 29
- “Identity Mapping Icons” on page 30
- “Supported Browsers” on page 31
- “Command Line Interface (CLI)” on page 32
- “Working with CLI Scripting” on page 40

Oracle ZFS Storage Appliance Key Features
Supported Protocols

The Oracle ZFS Storage Appliance includes technologies to deliver the best storage price/performance and unprecedented observability of your workloads in production, including:

- Analytics, a system for dynamically observing the behavior of your system in real-time and viewing data graphically
- The ZFS Hybrid Storage Pool, composed of optional Flash-memory devices for acceleration of reads and writes, low-power, high-capacity disks, and DRAM memory, all managed transparently as a single data hierarchy
- Support for a variety of hardware

For more information about Analytics and Hardware, refer to the documentation on the Oracle Technology Network (http://www.oracle.com/technetwork/indexes/documentation/index.html)

Supported Protocols

The Oracle ZFS Storage Appliance supports a variety of industry-standard client protocols, including NFS, iSCSI, SMB, FTP, HTTP, NDMP, Fibre Channel, SRP, iSER, and SFTP. For more information on these protocols, see:

- “Configuring SAN Fibre Channel” on page 104
- “Configuring SAN iSER Targets” on page 116
- “Working with NFS” on page 183
- “Working with iSCSI” on page 187
- “Working with SMB” on page 189
- “Working with FTP” on page 205
- “Working with HTTP” on page 206
- “Working with NDMP” on page 208
- “Working with SFTP” on page 216
- “Working with SRP” on page 219

Oracle ZFS Storage Appliance Data Services

To manage the data that you export using these protocols, you can configure the appliance using the built-in collection of advanced data services, including:

LICENSE NOTICE: Remote Replication and Cloning may be evaluated free of charge, but each feature requires that an independent license be purchased separately for use in production. After the evaluation period, these features must either be licensed or deactivated. Oracle reserves the right to audit for licensing compliance at any time. For details, refer to the "Oracle Software License Agreement ("SLA") and Entitlement for Hardware Systems with Integrated Software Options."
Data Availability

To maximize the availability of your data in production, the appliance includes a complete end-to-end architecture for data integrity, including redundancies at every level of the stack. Key features include:

- Predictive self-healing and diagnosis of all system hardware failures: CPUs, DRAM, I/O cards, disks, fans, power supplies
- ZFS end-to-end data checksums of all data and metadata, protecting data throughout the stack
- RAID-6 (double- and triple-parity) and optional RAID-6 across disk shelves
- Active-active clustering for high availability (See “Configuring Clustering” on page 141)
- Link aggregations and IP multipathing for network failure protection (See “Network Configuration” on page 56)
- I/O Multipathing between the controller and disk shelves
- Integrated software restart of all system software services (See “Working with Services”)
- Phone Home of telemetry for all software and hardware issues (See “Working with Phone Home” on page 245)
- Lights-out Management of each system for remote power control and console access

See Also:

- “Network Configuration” on page 56
- “Storage Configuration” on page 77
- “Configuring Storage Area Network” on page 99
Oracle ZFS Storage Appliance Configuration

To configure the appliance, use the following sections:

- “Initial Appliance Configuration” on page 49
- “Network Configuration” on page 56
- “Storage Configuration” on page 77
- “Configuring Storage Area Network” on page 99
- “Configuring Users” on page 124
- “Setting Oracle ZFS Storage Appliance Preferences” on page 132
- “Configuring Alerts” on page 134
- “Configuring Clustering” on page 141
- “Services”
- “Shares”

Browser User Interface (BUI)

The Oracle ZFS Storage Appliance Browser User Interface (BUI) is the graphical tool for administration of the appliance. The BUI provides an intuitive environment for administration tasks, visualizing concepts, and analyzing performance data. The BUI provides an uncluttered environment for visualizing system behavior and identifying performance issues with the appliance.
Direct your browser to the system using either the IP address or host name you assigned to the NET-0 port during initial configuration as follows: https://ipaddress:215 or https://hostname:215. The login screen appears.

The online help linked in the top right of the BUI is context-sensitive. For every top-level and second-level screen in the BUI, the associated help page appears when you click the Help button.
Changing a filesystem's properties by moving it into another project using the Projects side panel.

The masthead contains several interface elements for navigation and notification, as well as primary functionality. At left, from top to bottom, are the Sun/Oracle logo, a hardware model badge, and hardware power off/restart button. Across the right, again from top to bottom: login identification, logout, help, main navigation, and subnavigation.

System alerts appear in the Masthead as they are triggered. If multiple alerts are triggered sequentially, refer to the list of recent alerts found on the Dashboard screen or the full log available on the Logs screen.
Use the main navigation links to view between the Configuration, Maintenance, Shares, Status, and Analytics areas of the BUI. Use sub-navigation links to access features and functions within each area.

If you provide a session annotation, it appears beneath your login ID and the logout control. To change your session annotation for subsequent administrative actions without logging out, click on the text link. For details about session annotations, see “Configuring Users” on page 124.

The title bar appears below the Masthead and provides local navigation and functions that vary depending on the current view.

For example, the Identity mapping service title bar enables the following:
- Navigation to the full list of services through the side panel
- Controls to enable or disable the Identity Mapping service
- A view of Identity Mapping uptime
- Navigation to the Properties, Rules and Logs screens for your Identity Mapping service
- Button to Apply configuration changes made on the current screen
- Button to Revert configuration changes applied on the current screen

To quickly navigate between Service and Project views, open and close the side panel by clicking the title or the reveal arrow.
To add projects, click the Add... link in the sidebar.

To move Shares between Projects, click the move icon and drag a filesystem Share to the appropriate Project in the side panel.

Note that dragging a share into another project will change its properties if they are set to be inherited from its parent project.

Most BUI controls use standard web form inputs; however, there are a few key exceptions worth noting:

**TABLE 1**  
**Key Web Form Exceptions**

<table>
<thead>
<tr>
<th>Summary of BUI Controls</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify a property</td>
<td>Click the edit icon and complete the dialog</td>
</tr>
<tr>
<td>Add a list item or property entry</td>
<td>Click the add icon</td>
</tr>
<tr>
<td>Remove a list item or property entry</td>
<td>Click the remove icon</td>
</tr>
<tr>
<td>Save changes</td>
<td>Click the Apply button</td>
</tr>
<tr>
<td>Undo saved changes</td>
<td>Click the Revert button</td>
</tr>
<tr>
<td>Delete an item from a list</td>
<td>Click the trash icon (hover the mouse over the item row to see the icon)</td>
</tr>
<tr>
<td>Search for an item in a list</td>
<td>Click the search icon at the top right of the list</td>
</tr>
<tr>
<td>Sort by list headings</td>
<td>Click on the bold sub-headings to re-sort the list</td>
</tr>
<tr>
<td>Move or drag an item</td>
<td>Click the move icon</td>
</tr>
<tr>
<td>Rename an item</td>
<td>Click the rename icon</td>
</tr>
<tr>
<td>View details about your system</td>
<td>Oracle logo or click the model badge to go to the oracle.com web page for your model</td>
</tr>
<tr>
<td>Automatically open side panel</td>
<td>Drag an item to the side panel</td>
</tr>
</tbody>
</table>

When setting permissions, the RWX boxes are clickable targets. Clicking on the access group label (User, Group, Other) toggles all permissions for that label on and off.
To edit Share properties, deselect the Inherit from project checkbox.

![Property settings](image)

To view controls for an item in a list, hover the mouse over the row.

![Filesystem list](image)

All modal dialogs have titles and buttons that identify and commit or cancel the current action at top, and content below. The modal content area follows the same interface conventions as the main content area, but are different in that they must be dismissed using the buttons in the title bar before other actions can be performed.
Icons indicate system status and provide access to functionality, and in most cases serve as buttons to perform actions when clicked. It is useful to hover your mouse over interface icons to view the tooltip. The tables below provide a key to the conventions of the user interface.

The status lights are basic indicators of system health and service state:

**TABLE 2** Status Indicators

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Green on icon]</td>
<td>on</td>
<td>![Yellow warning icon]</td>
<td>warning</td>
</tr>
<tr>
<td>![Blue off icon]</td>
<td>off</td>
<td>![Red disabled icon]</td>
<td>disabled</td>
</tr>
</tbody>
</table>

The following icons are found throughout the user interface and cover most of the basic functionality:

**TABLE 3** BUI Icons

<table>
<thead>
<tr>
<th>Icon*</th>
<th>Description</th>
<th>Icon*</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Rename icon]</td>
<td>rename (edit text)</td>
<td>![Sever icon]</td>
<td>sever</td>
</tr>
<tr>
<td>![Move icon]</td>
<td>move</td>
<td>![Clone icon]</td>
<td>clone</td>
</tr>
<tr>
<td>![Edit icon]</td>
<td>edit</td>
<td>![Rollback icon]</td>
<td>rollback</td>
</tr>
<tr>
<td>![Destroy icon]</td>
<td>destroy</td>
<td></td>
<td>appliance power</td>
</tr>
</tbody>
</table>
### Browser User Interface (BUI)

#### About the Oracle ZFS Storage Appliance

The following icons are used to distinguish different types of objects and provide information of secondary importance.

#### TABLE 4  Miscellaneous Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Icon" /></td>
<td>allow</td>
<td><img src="image2" alt="Icon" /></td>
<td>SAS</td>
</tr>
<tr>
<td><img src="image3" alt="Icon" /></td>
<td>deny</td>
<td><img src="image4" alt="Icon" /></td>
<td>SAS port</td>
</tr>
<tr>
<td><img src="image5" alt="Icon" /></td>
<td>storage pool</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Disabled icons are shown at left.

See Also:
- “Initial Appliance Configuration” on page 49
- “Network Configuration” on page 56
- “Storage Configuration” on page 77
- “Configuring Storage Area Network” on page 99
Network Icons

These icons indicate the state of network devices and type of network datalinks:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>active network device</td>
<td>active Infiniband port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inactive network device</td>
<td>inactive Infiniband port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>network datalink</td>
<td>network datalink (IB partition)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>network datalink VLAN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>network datalink aggregation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>network datalink aggregation VLAN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See Also:
- “Initial Appliance Configuration” on page 49
- “Network Configuration” on page 56
- “Storage Configuration” on page 77
- “Configuring Storage Area Network” on page 99
- “Configuring Users” on page 124
- “Setting Oracle ZFS Storage Appliance Preferences” on page 132

Dashboard Icons

The following icons indicate the current state of monitored statistics with respect to user-configurable thresholds set from within Settings.
TABLE 6  Dashboard Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>☀</td>
<td>sunny</td>
<td>⚡</td>
<td>hurricane</td>
</tr>
<tr>
<td>☁</td>
<td>partly cloudy</td>
<td>⚡</td>
<td>hurricane class 2</td>
</tr>
<tr>
<td>☁</td>
<td>cloudy</td>
<td>⚡</td>
<td>hurricane class 3</td>
</tr>
<tr>
<td>🌧️</td>
<td>rainy</td>
<td>⚡</td>
<td>hurricane class 4</td>
</tr>
<tr>
<td>⚡</td>
<td>stormy</td>
<td>⚡</td>
<td>hurricane class 5</td>
</tr>
</tbody>
</table>

See Also:
- “Understanding the Appliance Status” on page 86
- “Status Dashboard Settings” on page 95
- “Initial Appliance Configuration” on page 49
- “Network Configuration” on page 56
- “Storage Configuration” on page 77
- “Configuring Storage Area Network” on page 99
- “Configuring Alerts” on page 134
- “Services”
- “Shares”

Analytics Toolbar Icons

This set of icons is used in a toolbar to manipulate display of information within Analytics worksheets.

TABLE 7  Analytics Toolbar Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>⬅️</td>
<td>back</td>
<td>⬆️</td>
<td>show minimum</td>
</tr>
<tr>
<td>⬅️</td>
<td>forward</td>
<td>⬆️</td>
<td>show maximum</td>
</tr>
<tr>
<td>⬅️</td>
<td>forward to now</td>
<td>⬆️</td>
<td>show line graph</td>
</tr>
</tbody>
</table>
Identity Mapping Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pause</td>
<td>show mountain graph</td>
<td>show one</td>
<td>unsync worksheet statistics</td>
</tr>
<tr>
<td>zoom out</td>
<td>crop outliers</td>
<td>one minute</td>
<td>unsync worksheet statistics</td>
</tr>
<tr>
<td>zoom in</td>
<td>sync worksheet to this statistic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>show one</td>
<td>drilldown</td>
<td>hour</td>
<td></td>
</tr>
<tr>
<td>show one</td>
<td>export statistical data</td>
<td>day</td>
<td>save statistical data</td>
</tr>
<tr>
<td>show one</td>
<td></td>
<td>week</td>
<td>archive dataset</td>
</tr>
<tr>
<td>show one</td>
<td></td>
<td>month</td>
<td>send worksheet with support bundle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See Also:
- “Understanding the Appliance Status” on page 86
- “Network Configuration” on page 56
- “Storage Configuration” on page 77
- “Configuring Alerts” on page 134
- “Services”

For more information about Analytics, refer to the documentation on the Oracle Technology Network (http://www.oracle.com/technetwork/indexes/documentation/index.html)

Identity Mapping Icons

These icons indicate the type of role being applied when mapping users and groups between Windows and Unix.

TABLE 8 Identity Mapping Icons

<table>
<thead>
<tr>
<th>Icon*</th>
<th>Description</th>
<th>Icon*</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄►►</td>
<td>allow Windows to Unix</td>
<td>◀◂◂</td>
<td>allow Unix to Windows</td>
</tr>
</tbody>
</table>
Supported Browsers

This section defines BUI browser support. For best results, use a tier 1 browser.

The BUI software is designed to be fully featured and functional on the following tier 1 browsers:

- Firefox 3.x or later
- Internet Explorer 7 or later
- Safari 3.1 or later
- Google Chrome (Stable)
- WebKit 525.13 or later

BUI elements may be cosmetically imperfect in tier 2 browsers, and some functionality may not be available, although all necessary features work correctly. A warning message appears during login if you are using one of the following tier 2 browsers:

- Firefox 2.x
- Mozilla 1.7 on Solaris 10
- Opera 9

Internet Explorer 6 and earlier versions are unsupported and known to have issues; login will not complete.

See Also:

- “Configuring Users” on page 124
Command Line Interface (CLI)

The CLI is designed to mirror the capabilities of the BUI, while also providing a powerful scripting environment for performing repetitive tasks. The command line is an efficient and powerful tool for repetitive administrative tasks. The appliance presents a CLI available through either the Console or SSH. There are several situations in which the preferred interaction with the system is the CLI:

- **Network unavailability** - If the network is unavailable, browser-based management is impossible; the only vector for management is the Console, which can only accommodate a text-based interface
- **Expediency** - Starting a browser may be prohibitively time-consuming, especially if you only want to examine a particular aspect of the system or make a quick configuration change
- **Precision** - In some situations, the information provided by the browser may be more qualitative than quantitative in nature, and you need a more precise answer
- **Automation** - Browser-based interaction cannot be easily automated; if you have repetitive or rigidly defined tasks, script the tasks

When navigating through the CLI, there are two principles to be aware of:

- **Tab completion is used extensively** - if you are not sure what to type in any given context, pressing the Tab key will provide you with possible options. Throughout the documentation, pressing Tab is presented as the word "tab" in bold italics.
- **Help is always available** - the *help* command provides context-specific help. Help on a particular topic is available by specifying the topic as an argument to *help*, for example *help commands*. Available topics are displayed by tab-completing the *help* command, or by typing *help topics*.

You can combine these two principles as follows:

```
dory:> help tab
builtins commands general help properties script
```

To log in remotely using the CLI, use an *ssh* client. If you have not followed the instructions in “Configuring Users” on page 124 to administer the appliance, you will need to log in as root. When you log in, the CLI will present you with a prompt that consists of the hostname, followed by a colon, followed by a greater-than sign:

```
% ssh root@dory
Password:
Last login: Mon Oct 13 15:43:05 2009 from kiowa.sf.fishpo
dory:> 
```

See Also:
CLI Contexts

A central principle in the CLI is the context in which commands are executed. The context dictates which elements of the system can be managed and which commands are available. Contexts have a tree structure in which contexts may themselves contain nested contexts and the structure generally mirrors that of the views in the BUI.

The initial context upon login is the root context, and serves as the parent or ancestor of all contexts. To navigate to a context, execute the name of the context as a command. For example, the functionality available in the Configuration view in the browser is available in the configuration context of the CLI. From the root context, this can be accessed by typing it directly:

dory:> configuration
dory:configuration>

Note that the prompt changes to reflect the context, with the context provided between the colon and the greater-than sign in the prompt.

The show command shows child contexts. For example, from the configuration context:

dory:configuration> show
Children:
  net => Configure networking
  services => Configure services
  version => Display system version
  users => Configure administrative users
  roles => Configure administrative roles
  preferences => Configure user preferences
  alerts => Configure alerts
  storage => Configure Storage

These child contexts correspond to the views available under the Configuration view in the browser, including Network, Services and Users, Preferences, and so on. To select one of these child contexts, type its name:

dory:configuration> preferences
dory:configuration preferences>

Navigate to a descendant context directly from an ancestor by specifying the intermediate contexts separated with spaces. For example, to navigate directly to configuration preferences from the root context, simply type it:

dory:> configuration preferences
Some child contexts are *dynamic* in that they correspond not to fixed views in the browser, but rather to dynamic entities that have been created by either the user or the system. To navigate to these contexts, use the `select` command, followed by the name of the dynamic context. The names of the dynamic contexts contained within a given context are shown using the `list` command. For example, the `users` context is a static context, but each user is its own dynamic context.

```
dory:configuration users
```

```
dory:configuration users> list
NAME    USERNAME  UID  TYPE
John Doe bmc   12345  Dir
Super-User root 0  Loc
```

To select the user named `bmc`, issue the command `select bmc`:

```
dory:configuration users select bmc
```

Alternately, `select` and `destroy` can in some contexts be used to select an entity based on its properties. For example, one could select log entries issued by the `reboot` module in the maintenance logs system context by issuing the following command:

```
dory:maintenance logs system> select module=reboot
```

```
dory:maintenance logs system entry-034> show
Properties:
  timestamp = 2010-8-14 06:24:41
  module = reboot
  priority = crit
  text = initiated by root on /dev/console syslogd: going down on signal 15
```

As with other commands, `select` may be appended to a context-changing command. For example, to select the user named `bmc` from the root context:

```
dory:> configuration users select bmc
```

Use the `last` command to navigate to a previously selected or created context. The following example creates a replication action, and then uses the `last` and `get id` commands to retrieve the replication action ID. Then a different action is selected, and the `last` and `get id` commands are used to retrieve the ID of the last-visited replication action.

```
dory:configuration net interfaces> select igb4
```

```
dory:configuration net interfaces igb4> done
```

```
dory:configuration net interfaces> last
```

```
et:configuration net interfaces igb4>
```

The `last` command is also useful to retrieve values that have been automatically set by the appliance during the creation of a dynamic node. For example, each replication action is
assigned an ID by the appliance when it is created. Using the `last` command with the `get id` command, you can retrieve the ID without using the name of the replication action:

```
dory:shares pl/share replication> create
dory:shares pl/share action (uncommitted)> set target=dory
target = dory (uncommitted)
dory:shares pl/share action (uncommitted)> set pool=p0
pool = p0 (uncommitted)
dory:shares pl/share action (uncommitted)> commit
```

```
dory:shares pl/share replication> last get id
id = 7034367a-d4d8-e26f-fa93-c3b454e3b595
dory:shares pl/share replication>
```

Note that when `last` is combined with another command (in this case, `get id`), the command is run in the context of the last-visited node, but the current node remains unchanged.

Because `last` allows you to retrieve the last-visited node and its values without specifying the name of the node, this command is particularly convenient for scripting:

```
script
    project = 'myproj';
target = 'mytarget';
target_pool = 'notmypool';

    run('cd /');
    run('shares select ' + project);
    run('replication');
    run('create');
    set('target', target);
    set('pool', target_pool);
    run('commit');
    run('last');
    id = get('id');
    printf("Sending update for replication action id %s ...", id);
    run('sendupdate');
    while (get('state') != 'idle') {
        printf(".");
        run('sleep 1');
    }
    printf("done
");
```

To return to the previous context, use the `done` command:

```
dory:configuration> done
dory:>
```

This returns to the previous context, which is not necessarily the parent context, as follows:

```
dory:配置> configuration users select bmc
dory:configuration users bmc> done
```

The `done` command can be used multiple times to backtrack to earlier contexts:
To navigate to a parent context, use the `cd` command. Inspired by the classic UNIX command, `cd` takes an argument of ".." to denote moving to the parent context:

```
dory:> configuration users select bmc
dory:configuration users bmc> cd..
dory:configuration users>
```

And as with the UNIX command, "cd /" moves to the root context:

```
dory:> configuration users select bmc
dory:configuration users bmc> cd /
dory:>
```

And as with its UNIX analogue, "cd ../.." may be used to navigate to the grandparent context:

```
dory:> configuration users select bmc
dory:configuration users bmc> cd ../..
dory:configuration>
```

Context names will tab complete, be they static contexts (via normal command completion) or dynamic contexts (via command completion of the `select` command). Following is an example of selecting the user named `bmc` from the root context with just fifteen keystrokes, instead of the thirty-one that would be required without tab completion:

```
dory:> configtab
dory:> configuration utab
dory:> configuration users setab
dory:> configuration users select tab
bmc root
```

Once in a context, execute context-specific commands. For example, to get the current user's preferences, execute the `get` command from the `configuration preferences` context:

```
dory:configuration preferences> get
    locale = C
    login_screen = status/dashboard
```
The CLI provides commands to create and manage resources in the Oracle ZFS Storage Appliance. For example, to get preferences from the root context without changing context, append the `get` command to the context navigation commands:

```
dory:> configuration preferences get
locale = C
login_screen = status/dashboard
session_timeout = 15
session_annotation =
advanced_analytics = false
```

When creating a new entity in the system, the context associated with the new entity will often be created in an *uncommitted* state. For example, create a threshold alert by executing the `create` command from the `configuration alerts threshold` context:

```
dory:> configuration alerts thresholds create
dory:configuration alerts threshold (uncommitted)>
```

The `(uncommitted)` in the prompt denotes that this is an uncommitted context. An uncommitted entity is committed via the `commit` command; any attempt to navigate away from the uncommitted context will prompt for confirmation:

```
dory:configuration alerts threshold (uncommitted)> cd /
Leaving will abort creation of "threshold". Are you sure? (Y/N)
```

When committing an uncommitted entity, the properties associated with the new entity will be validated, and an error will be generated if the entity cannot be created. For example, the creation of a new threshold alert requires the specification of a statistic name; failure to set this name results in an error:

```
dory:configuration alerts threshold (uncommitted)> commit
error: missing value for property "statname"
```

To resolve the problem, address the error and reattempt the `commit`:

```
dory:configuration alerts threshold (uncommitted)> set statname=cpu.utilization
statname = cpu.utilization (uncommitted)
dory:configuration alerts threshold (uncommitted)> commit
error: missing value for property "limit"
```

```
dory:configuration alerts threshold (uncommitted)> set limit=90
limit = 90 (uncommitted)
dory:configuration alerts threshold (uncommitted)> commit
```

```
dory:configuration alerts thresholds> list
THRESHOLD          LIMIT       TYPE STATNAME
threshold-000         90     normal cpu.utilization
```

See Also:
CLI Properties

Properties are typed name/value pairs that are associated with a context. Properties for a given context can be ascertained by running the "help properties" command. Following is an example of retrieving the properties associated with a user's preferences:

```
dory:configuration preferences> help properties
Properties that are valid in this context:
locale               => Locality
login_screen         => Initial login screen
session_timeout      => Session timeout
session_annotation   => Current session annotation
advanced_analytics   => Make available advanced analytics statistics
```

The properties of a given context can be retrieved with the get command. Following is an example of using the get command to retrieve a user's preferences:

```
dory:configuration preferences> get
locale = C
login_screen = status/dashboard
session_timeout = 15
session_annotation =
advanced_analytics = false
```

The get command will return any properties provided to it as arguments. For example, to get the value of the login_screen property:

```
dory:configuration preferences> get login_screen
login_screen = status/dashboard
```

The get command will tab complete with the names of the available properties. For example, to see a list of available properties for the iSCSI service:

```
dory:> configuration services iscsi get tab
<status>       isns_server   radius_secret   target_chap_name
isns_access    radius_access radius_server target_chap_secret
```

The set command will set a property to a specified value, with the property name and its value separated by an equals sign. For example, to set the login_screen property to "shares":

```
dory:configuration preferences> set login_screen=shares
login_screen = shares (uncommitted)
```
Note that in the case of properties that constitute state on the appliance, setting the property does not change the value, but rather records the set value and indicates that the value of the property is uncommitted.

To force set property values to take effect, they must be explicitly committed, allowing multiple values to be changed as a single, coherent change. To commit any uncommitted property values, use the `commit` command:

```
dory:configuration preferences> get login_screen
    login_screen = shares (uncommitted)
dory:configuration preferences> commit

dory:configuration preferences> get login_screen
    login_screen = shares
```

If you attempt to leave a context that contains uncommitted properties, you will be warned that leaving will abandon the set property values, and will be prompted to confirm that you wish to leave. For example:

```
dory:configuration preferences> set login_screen=maintenance/hardware
    login_screen = maintenance/hardware (uncommitted)
dory:configuration preferences> done
You have uncommitted changes that will be discarded. Are you sure? (Y/N)
```

If a property in a context is set from a different context -- that is, if the `set` command has been appended to a command that changes context -- the commit is implied, and happens before control is returned to the originating context. For example:

```
dory:> configuration preferences set login_screen=analytics/worksheets
    login_screen = analytics/worksheets
```

Some properties take a list of values. For these properties, the list elements should be separated by a comma. For example, the NTP servers property may be set to a list of NTP servers:

```
dory:configuration services ntp> set servers=0.pool.ntp.org,1.pool.ntp.org
    servers = 0.pool.ntp.org,1.pool.ntp.org (uncommitted)
dory:configuration services ntp> commit
```

If a property value contains a comma, an equals sign, a quote or a space, the entire value must be quoted. For example, to set the `sharenfs` shares property for the default project to be read-only but provide read/write access to the host "kiowa". For more information, see “Working with Shares”.

```
dory:> shares select default

dory:shares default> set sharenfs="ro,rw=kiowa"
    sharenfs = ro,rw=kiowa (uncommitted)
dory:shares default> commit
```

Some properties are immutable; you can get their values, but you cannot set them. Attempts to set an immutable property results in an error. For example, attempting to set the immutable space_available property of the default project. For more information, see “Working with Shares”.
dory:> shares select default
dory:shares default> get space_available
   space_available = 1.15T
dory:shares default> set space_available=100P
   error: cannot set immutable property "space_available"

Some other properties are only immutable in certain conditions. For these properties, the set command is not valid. For example, if the user named bmc is a network user, the fullname property will be immutable:

dory:> configuration users select bmc set fullname="Rembrandt Q. Einstein"
   error: cannot set immutable property "fullname"

See Also:
- “Browser User Interface (BUI)” on page 20
- “Command Line Interface (CLI)” on page 32

---

**Working with CLI Scripting**

The CLI is designed to provide a powerful scripting environment for performing repetitive tasks.

You can use Batching Commands or Scripting Commands (or some combination), but in any case the automated infrastructure requires automated access to the appliance. This must be done by user configuration, user authorizations, and setting SSH public keys using the CLI. For more information, see:

- “Configuring Users” on page 124
- “User Authorizations” on page 125
- “Setting SSH Public Keys using the CLI” on page 134

To use CLI scripting, use the following sections:

- “Using Batch Commands” on page 41
- “Understanding the CLI Scripting Commands” on page 41
- “Accessing the CLI Script Environment” on page 42
- “Understanding the Built-in CLI Functions” on page 42
- “Using the Run Function” on page 43
- “Using the Get Function” on page 43
- “Using the List Function” on page 44
- “Using the Children Function” on page 45
- “Using the Choices Function” on page 46
- “Using the Functions for Generating Output” on page 47
- “Understanding CLI Scripting Errors” on page 47
Using Batch Commands

The simplest scripting mechanism is to batch appliance shell commands. For example, to automatically take a snapshot called "newsnap" in the project "myproj" and the filesystem "myfs", put the following commands in a file:

```sh
shares
select myproj
select myfs
snapshots snapshot newsnap
```

Then ssh onto the appliance, redirecting standard input to be the file:

```sh%
ssh root@dory < myfile.txt
```

In many shells, you can abbreviate this by using a "here file", where input up to a token is sent to standard input. Following is the above example in terms of a here file:

```sh%
```

```

This mechanism is sufficient for the simplest kind of automation, and may be sufficient if wrapped in programmatic logic in a higher-level shell scripting language on a client, but it generally leaves much to be desired.

Understanding the CLI Scripting Commands

While batching commands is sufficient for the simplest of operations, it can be tedious to wrap in programmatic logic. For example, if you want to get information on the space usage for every share, you must have many different invocations of the CLI, wrapped in a higher level language on the client that parsed the output of specific commands. This results in slow, brittle automation infrastructure. To allow for faster and most robust automation, the appliance has a rich scripting environment based on ECMAScript 3. An ECMAScript tutorial is beyond the scope of this document, but it is a dynamically typed language with a C-like syntax that allows for:

- Conditional code flow (if/else)
- Iterative code flow (while, for, etc.)
- Structural and array data manipulation via first-class Object and Array types
- Perl-like regular expressions and string manipulation (split(), join(), etc.)
- Exceptions
- Sophisticated functional language features like closures
# Accessing the CLI Script Environment

1. In the CLI, enter the script environment using the `script` command:

   ```
   dory:> script
   ("." to run)>
   ```

2. At the script environment prompt, you can input your script, finally entering "." alone on a line to execute it:

   ```
   dory:> script
   ("." to run)> for (i = 10; i > 0; i--)
   ("." to run)>   printf("%d... ", i);
   ("." to run)>   printf("Blastoff!\n");
   ("." to run)> .
   10... 9... 8... 7... 6... 5... 4... 3... 2... 1... Blastoff!
   ```

3. If your script is a single line, you can simply provide it as an argument to the `script` command, making for an easy way to explore scripting:

   ```
   dory:> script print("It is now " + new Date())
   It is now Tue Oct 14 2009 05:33:01 GMT+0000 (UTC)
   ```

## Understanding the Built-in CLI Functions

Of course, scripts are of little utility unless they can interact with the system at large. There are several built-in functions that allow your scripts to interact with the system:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>get</code></td>
<td>Gets the value of the specified property. Note that this function returns the value in native form, e.g., dates are returned as Date objects.</td>
</tr>
<tr>
<td><code>list</code></td>
<td>Returns an array of tokens corresponding to the dynamic children of the current context.</td>
</tr>
<tr>
<td><code>run</code></td>
<td>Runs the specified command in the shell, returning any output as a string. Note that if the output contains multiple lines, the returned string will contain embedded newlines.</td>
</tr>
<tr>
<td><code>props</code></td>
<td>Returns an array of the property names for the current node.</td>
</tr>
<tr>
<td><code>set</code></td>
<td>Takes two string arguments, setting the specified property to the specified value.</td>
</tr>
<tr>
<td><code>choices</code></td>
<td>Returns an array of the valid property values for any property for which the set of values is known and enumerable.</td>
</tr>
</tbody>
</table>
Using the Run Function

1. The simplest way for scripts to interact with the larger system is to use the "run" function: it takes a command to run, and returns the output of that command as a string. For example:

   dory:> configuration version script dump(run('get boot_time'))
   boot_time = 2009-10-12 07:02:17

2. The built-in `dump` function dumps the argument out, without expanding any embedded newlines. ECMAScript's string handling facilities can be used to take apart output. For example, splitting the above based on whitespace:

   dory:> configuration version script dump(run('get boot_time').split(/\s+/))
   ['', 'boot_time', '=', '2009-10-12', '07:02:17', '']

Using the Get Function

The `run` function is sufficiently powerful that it may be tempting to rely exclusively on parsing output to get information about the system -- but this has the decided disadvantage that it leaves scripts parsing human-readable output that may or may not change in the future. To more robustly gather information about the system, use the built-in "get" function. In the case of the `boot_time` property, this will return not the string but rather the ECMAScript `Date` object, allowing the property value to be manipulated programmatically.

1. For example, you might want to use the `boot_time` property in conjunction with the current time to determine the time since boot:

   ```javascript
   script
   run('configuration version');
   now = new Date();
   uptime = (now.valueOf() - get('boot_time').valueOf()) / 1000;
   printf('up %d day%s, %d hour%s, %d minute%s, %d second%s
',
      d = uptime / 86400, d < 1 || d >= 2 ? 's' : '',
      h = (uptime / 3600) % 24, h < 1 || h >= 2 ? 's': '',
      m = (uptime / 60) % 60, m < 1 || m >= 2 ? 's': '',
      s = uptime % 60, s < 1 || s >= 2 ? 's': '');
   ```

2. Assuming the above is saved as a "uptime.aksh", you could run it this way:

   ```bash
   % ssh root@dory < uptime.aksh
   Pseudo-terminal will not be allocated because stdin is not a terminal.
   Password:
   up 2 days, 10 hours, 47 minutes, 48 seconds
   ```

   The message about pseudo-terminal allocation is due to the ssh client; the issue that this message refers to can be dealt with by specifying the "-T" option to ssh.
Using the List Function

1. In a context with dynamic children, it can be very useful to iterate over those children programmatically. This can be done by using the `list` function, which returns an array of dynamic children. For example, following is a script that iterates over every share in every project, printing out the amount of space consumed and space available:

   ```
   script
     run('shares');
     projects = list();
     for (i = 0; i < projects.length; i++) {
       run('select ' + projects[i]);
       shares = list();
       for (j = 0; j < shares.length; j++) {
         run('select ' + shares[j]);
         printf("%s/%s %1.64g %1.64g\n", projects[i], shares[j], get('space_data'), get('space_available'));
       }
     }
   run('cd ..');
   ```

2. Here is the output of running the script, assuming it were saved to a file named "space.aksh":

   ```
   % ssh root@koi < space.aksh
   Password:
   admin/accounts 18432 266617007104
   admin/exports 18432 266617007104
   admin/primary 18432 266617007104
   admin/traffic 18432 266617007104
   admin/workflow 18432 266617007104
   alevanthal/hw_eng 18432 266617007104
   bcantrill/analytix 1073964032 266617007104
   bgregg/dashbd 18432 266617007104
   bgregg/filesys01 26112 107374156288
   bpiljewski/access_ctrl 18432 266617007104
   ...
   ```

3. If one would rather a "pretty printed" (though more difficult to handle programmatically) variant of this, one could directly parse the output of the `get` command:

   ```
   script
     run('shares');
     projects = list();
     printf("%-40s %-10s %-10s\n", 'SHARE', 'USED', 'AVAILABLE');
   ```
Using the Children Function

Even in a context with static children, it can be useful to iterate over those children programmatically. This can be done by using the `children` function, which returns an array of static children.

1. **For example, here’s a script that iterates over every service, printing out the status of the service:**

   ```
   configuration services
   script
   var svcs = children();
   for (var i = 0; i < svcs.length; ++i) {
   ```

4. **And here is some of the output of running this new script, assuming it were named “prettyspace.aksh”:**

   ```
   % ssh root@koi < prettyspace.aksh
   Password:
   SHARE                                    USED       AVAILABLE
   admin/accounts                           18K        248G
   admin/exports                            18K        248G
   admin/primary                            18K        248G
   admin/traffic                            18K        248G
   admin/workflow                           18K        248G
   aleventhal/hw_eng                        18K        248G
   bcantrill/analytx                        1.00G      248G
   bgregg/dashbd                            18K        248G
   bgregg/filesys01                         25.5K      100G
   bpijewski/access_ctrl                    18K        248G
   ...
Using the Choices Function

The choices function returns an array of the valid property values for any property for which the set of values is known and enumerable. For example, the following script retrieves the list of all pools on the shares node using the choices function and then iterates all pools to list projects and shares along with the available space.

1. For example, the following script retrieves the list of all pools on the shares node using the choices function and then iterates all pools to list projects and shares along with the available space.

```plaintext
fmt = '%-40s %-15s %-15s
';
printf(fmt, 'SHARE', 'USED', 'AVAILABLE');
rund('cd /');
rund('shares');
pools = choices('pool');
for (p = 0; p < pools.length; p++) {
```
set('pool', pools[p]);
projects = list();
for (i = 0; i < projects.length; i++) {
    run('select ' + projects[i]);
    shares = list();
    for (j = 0; j < shares.length; j++) {
        run('select ' + shares[j]);
        share = pools[p] + ':' + projects[i] + '/' + shares[j];
        printf(fmt, share, get('space_data'),
               get('space_available'));
        run('cd ..');
    }
    run('cd ..');
}

2. Here is the output of running the script:

<table>
<thead>
<tr>
<th>SHARE</th>
<th>USED</th>
<th>AVAILABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>pond:projectA/fs1</td>
<td>31744</td>
<td>566196178944</td>
</tr>
<tr>
<td>pond:projectA/fs2</td>
<td>31744</td>
<td>566196178944</td>
</tr>
<tr>
<td>pond:projectB/lun1</td>
<td>21474836480</td>
<td>587670999040</td>
</tr>
<tr>
<td>puddle:deptA/share1</td>
<td>238475</td>
<td>467539219283</td>
</tr>
<tr>
<td>puddle:deptB/share1</td>
<td>129564</td>
<td>467539219283</td>
</tr>
<tr>
<td>puddle:deptB/share2</td>
<td>19283747</td>
<td>467539219283</td>
</tr>
</tbody>
</table>

Using the Functions for Generating Output

Reporting state on the system requires generating output. Scripts have several built-in functions made available to them to generate output:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dump</td>
<td>Dumps the specified argument to the terminal, without expanding embedded newline. Objects will be displayed in a JSON-like format. Useful for debugging.</td>
</tr>
<tr>
<td>print</td>
<td>Prints the specified object as a string, followed by a newline. If the object does not have a toString method, it will be printed opaquely.</td>
</tr>
<tr>
<td>printf</td>
<td>Like C's printf(3C), prints the specified arguments according to the specified formatting string.</td>
</tr>
</tbody>
</table>

Understanding CLI Scripting Errors

When an error is generated, an exception is thrown. The exception is generally an object that contains the following members:
Using the Choices Function

- code - a numeric code associated with the error
- message - a human-readable message associated with the error

Exceptions can be caught and handled, or they may be thrown out of the script environment. If a script environment has an uncaught exception, the CLI will display the details. For example:

```
dory:> script run('not a cmd')
error: uncaught error exception (code EAKSH_BADCMD) in script: invalid command
   "not a cmd" (encountered while attempting to run command "not a cmd")
```

You could see more details about the exception by catching it and dumping it out:

```
dory:> script try { run('not a cmd') } catch (err) { dump(err); }
{
   toString: <function>,
   code: 10004,
   message: 'invalid command "not a cmd" (encountered while attempting to
           run command "not a cmd")'
}
```

This also allows you to have rich error handling, for example:

```
#!/usr/bin/ksh -p
ssh -T root@dory <<EOF
script
   try {
      run('shares select default select $1');
   } catch (err) {
      if (err.code == EAKSH_ENTITY_BADSELECT) {
         printf('error: "$1" is not a share in the ' +
                'default project\n');
         exit(1);
      }
      throw (err);
   }

   printf("default/$1": compression is %s\n', get('compression'));
exit(0);
EOF
```

If this script is named "share.ksh" and run with an invalid share name, a rich error message will be generated:

```
% ksh ./share.ksh bogus
error: "bogus" is not a share in the default project
```
Configuring the Oracle ZFS Storage Appliance

To configure the appliance, use the following sections:

- “Initial Appliance Configuration” on page 49
- “Network Configuration” on page 56
- “Storage Configuration” on page 77
- “Appliance Status” on page 86
- “Configuring Storage Area Network” on page 99
- “Configuring Users” on page 124
- “Setting Oracle ZFS Storage Appliance Preferences” on page 132
- “Configuring Alerts” on page 134
- “Configuring Clustering” on page 141

Initial Appliance Configuration

To perform the initial appliance configuration, use the following sections:

- “Initial Configuration using the BUI” on page 50
- “Initial Configuration using the CLI” on page 51

See Also:

- “Network Configuration” on page 56
- “DNS” on page 239
- “Time” on page 242
- Name Services - “NIS” on page 222, “LDAP” on page 224, “Active Directory” on page 228
- “Storage Configuration” on page 77
- “Registration & Support” on page 245

Perform initial system configuration after powering on the appliance for the first time and establishing a connection, as documented in the “Oracle ZFS Storage Appliance Installation Guide”.

The option to perform initial configuration of a cluster is only available in the BUI. If electing this option, read “Configuring Clustering” on page 141 before beginning initial configuration.
for detailed additional steps that are required for successful cluster setup. Pay careful attention to “Clustering Considerations for Networking” on page 154. Alternatively, cluster-capable appliances may be initially configured for standalone operation using the following procedure, and re-configured for cluster operation at a later time.

Initial configuration configures network connectivity, several client network services, and the storage pool layout for standalone operation. When completed, the appliance is ready for use but does no shares configured for remote clients to access. To create shares or revisit settings, see “Working with Shares”.

FIGURE 1 Oracle ZFS Storage Appliance Welcome Page

▼ Initial Configuration Using the BUI

Initial configuration can be repeated at a later time by clicking the "INITIAL SETUP" button on the System screen or by entering the maintenance system setup context in the CLI.

1. To start initial configuration, on the Welcome page click Start.

2. For each page to commit your changes and go to the next screen, click Commit.

3. To go to a previous screen use the arrow buttons.
Standalone and Clustered Controllers

Standalone controllers must have at least one NIC port configured as a management interface. Select the Allow Admin option in the BUI to enable BUI connections on port 215 and CLI connections on ssh port 22.

Cluster installations must have at least one NIC port on each controller configured as a management interface as described above. In addition, the NIC instance number must be unique on each controller.

Initial Configuration Using the CLI

You use the CLI to perform the initial configuration sections. Each step begins by printing its help, which can be reprinted by typing help. Use the done command to complete each step.

Login using the password you provided during installation:

    caji console login: root
    Password:
    Last login: Sun Oct 19 02:55:31 on console

To setup your system, you will be taken through a series of steps; as the setup process advances to each step, the help message for that step will be displayed.

Press any key to begin initial configuration ...

In this example, the existing settings are checked (which were obtained from the DHCP server), and accepted by typing done. To customize them at this point, enter each context (datalinks, devices and interfaces) and type help to see available actions for that context. See “Network Configuration” on page 56 for additional documentation. Pay careful attention to “Clustering Considerations for Networking” on page 154 if you will configure clustering.

    aksh: starting configuration with "net" ...

Configure Networking. Configure the appliance network interfaces. The first network interface has been configured for you, using the settings you provided at the serial console.

Subcommands that are valid in this context:

    datalinks => Manage datalinks
    devices   => Manage devices
    interfaces => Manage interfaces
help [topic] => Get context-sensitive help. If [topic] is specified, it must be one of "builtins", "commands", "general", "help" or "script".

show => Show information pertinent to the current context

abort => Abort this task (potentially resulting in a misconfigured system)

done => Finish operating on "net"

caji:maintenance system setup net> devices show
Devices:

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>UP</th>
<th>MAC</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>igb0</td>
<td>true</td>
<td>0:14:4f:8d:59:aa</td>
<td>1000 Mbit/s</td>
</tr>
<tr>
<td>igb1</td>
<td>false</td>
<td>0:14:4f:8d:59:ab</td>
<td>0 Mbit/s</td>
</tr>
<tr>
<td>igb2</td>
<td>false</td>
<td>0:14:4f:8d:59:ac</td>
<td>0 Mbit/s</td>
</tr>
<tr>
<td>igb3</td>
<td>false</td>
<td>0:14:4f:8d:59:ad</td>
<td>0 Mbit/s</td>
</tr>
</tbody>
</table>

caji:maintenance system setup net> datalinks show
Datalinks:

<table>
<thead>
<tr>
<th>DATALINK CLASS</th>
<th>LINKS</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>igb0 device</td>
<td>igb0</td>
<td>Untitled Datalink</td>
</tr>
</tbody>
</table>

caji:maintenance system setup net> interfaces show
Interfaces:

<table>
<thead>
<tr>
<th>INTERFACE</th>
<th>STATE</th>
<th>CLASS</th>
<th>LINKS</th>
<th>ADDRS</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>igb0</td>
<td>up</td>
<td>ip</td>
<td>igb0</td>
<td>192.168.2.80/22</td>
<td>Untitled Interface</td>
</tr>
</tbody>
</table>

caji:maintenance system setup net> done

Refer to “DNS” on page 239 for additional documentation about DNS.

Configure DNS. Configure the Domain Name Service.

Subcommands that are valid in this context:

help [topic] => Get context-sensitive help. If [topic] is specified, it must be one of "builtins", "commands", "general", "help", "script" or "properties".

show => Show information pertinent to the current context

commit => Commit current state, including any changes

abort => Abort this task (potentially resulting in a misconfigured system)

done => Finish operating on "dns"

get [prop] => Get value for property [prop]. ("help properties"
for valid properties.) If [prop] is not specified, returns values for all properties.

set [prop] => Set property [prop] to [value]. ("help properties" for valid properties.) For properties taking list values, [value] should be a comma-separated list of values.

caji:maintenance system setup dns> show
Properties:
  <status> = online
  domain = sun.com
  servers = 192.168.1.4

caji:maintenance system setup dns> set domain=sf.fishworks.com
  domain = sf.fishworks.com (uncommitted)
caji:maintenance system setup dns> set servers=192.168.1.5
  servers = 192.168.1.5 (uncommitted)
caji:maintenance system setup dns> commit
caji:maintenance system setup dns> done
aksh: done with "dns", advancing configuration to "ntp" ...

Configure Network Time Protocol (NTP) to synchronize the appliance time clock. See “NTP” on page 242 for additional documentation.

Configure Time. Configure the Network Time Protocol.

Subcommands that are valid in this context:

help [topic] => Get context-sensitive help. If [topic] is specified, it must be one of "builtins", "commands", "general", "help", "script" or "properties".

show => Show information pertinent to the current context

commit => Commit current state, including any changes

abort => Abort this task (potentially resulting in a misconfigured system)

done => Finish operating on "ntp"

enable => Enable the ntp service

disable => Disable the ntp service

get [prop] => Get value for property [prop]. ("help properties" for valid properties.) If [prop] is not specified, returns values for all properties.

set [prop] => Set property [prop] to [value]. ("help properties" for valid properties.) For properties taking list values, [value] should be a comma-separated list of
values.

caji:maintenance system setup ntp> set servers=0.pool.ntp.org
   servers = 0.pool.ntp.org (uncommitted)
caji:maintenance system setup ntp> commit
caji:maintenance system setup ntp> done
aksh: done with "ntp", advancing configuration to "directory" ...

For additional documentation, refer to:

■ “NIS” on page 222
■ “LDAP” on page 224
■ “Active Directory” on page 228

Configure Name Services. Configure directory services for users and groups. You can configure and enable each directory service independently, and you can configure more than one directory service.

Subcommands that are valid in this context:

nis $=> Configure NIS
ldap $=> Configure LDAP
ad $=> Configure Active Directory
help [topic] $=> Get context-sensitive help. If [topic] is specified, it must be one of "builtins", "commands", "general", "help" or "script".
show $=> Show information pertinent to the current context
abort $=> Abort this task (potentially resulting in a misconfigured system)
done $=> Finish operating on "directory"

caji:maintenance system setup directory> nis
caji:maintenance system setup directory nis> show
Properties:
   <status> = online
domain = sun.com
broadcast = true
ypservers =

caji:maintenance system setup directory nis> set domain=fishworks
domain = fishworks (uncommitted)
caji:maintenance system setup directory nis> commit
caji:maintenance system setup directory nis> done
caji:maintenance system setup directory> done
aksh: done with "directory", advancing configuration to "support" ...
Configure storage pools that are characterized by their underlying data redundancy, and provide space that is shared across all filesystems and LUNs. For additional documentation, see “Storage Configuration” on page 77.

Configure Storage.

Subcommands that are valid in this context:

```
help [topic] => Get context-sensitive help. If [topic] is specified, it must be one of "builtins", "commands", "general", "help", "script" or "properties".
show => Show information pertinent to the current context
commit => Commit current state, including any changes
done => Finish operating on "storage"
config <pool> => Configure the storage pool
unconfig => Unconfigure the storage pool
add => Add additional storage to the storage pool
import => Search for existing or destroyed pools to import
scrub <start|stop> => Start or stop a scrub
get [prop] => Get value for property [prop]. ("help properties" for valid properties.) If [prop] is not specified, returns values for all properties.
set pool=[pool] => Change current pool
```

```
caji:maintenance system setup storage> show
Properties:
   pool = pool-0
   status = online
   profile = mirror
   log_profile = -
   cache_profile = -
caji:maintenance system setup storage> done
aksh: done with "storage", advancing configuration to "support" ...
```

For additional documentation of remote support configuration, refer to “Phone Home” on page 245.

Remote Support. Register your appliance and configure remote monitoring.

Subcommands that are valid in this context:

```
tags => Configure service tags
```
Network Configuration

The Networking Configuration features lets you create a variety of advanced networking setups using your physical network ports, including link-aggregations, virtual NICs (VNICs), virtual LANs (VLANs), and multipathing groups. You can then define any number of IPv4 and IPv6 addresses for these abstractions, for use in connecting to the various data services on the system.

There are four components to a system's network configuration:

- **Devices** - Physical network ports. These correspond to your physical network connections or IP on InfiniBand (IPoIB) partitions.
- **Datalinks** - The basic construct for sending and receiving packets. Datalinks may correspond 1:1 with a device (that is, with a physical network port) or IB Partition, or you may define Aggregation, VLAN and VNIC datalinks composed of other devices and datalinks.
- **Interface** - The basic construct for IP configuration and addressing. Each IP interface is associated with a single datalink, or is defined to be an IP MultiPathing (IPMP) group comprised of other interfaces.
- **Routing** - IP routing configuration. This controls how the system will direct IP packets.

To configure the network for the appliance, use the following sections:

- “Working with the Network Configuration BUI Page” on page 57
- “Configuring Network Devices” on page 58
- “Configuring Network Datalinks” on page 58
- “Configuring Network Interfaces” on page 60
- “Configuring Network IP Multipathing (IPMP)” on page 61
- “Configuring Network Performance and Availability” on page 63
- “Configuring Network Routing” on page 63
Working with the Network Configuration BUI Page

In the appliance model, network devices represent the available hardware; they have no configurable settings. Datalinks are a layer 2 entity and must be created to apply settings such as LACP to these network devices. Interfaces are a layer 3 entity containing the IP settings, which they make available via a datalink. This model has separated network interface settings into two parts: datalinks for layer 2 settings and interfaces for layer 3 settings.

![Network Configuration Window](image)

An example of a single IP address on a single port (common configuration) is:

**TABLE 11** Example - Single IP Address on a Single Port

<table>
<thead>
<tr>
<th>Devices</th>
<th>Datalink</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>igb0</td>
<td>datalink1</td>
<td>deimos (192.168.2.80/22)</td>
</tr>
</tbody>
</table>

The following configuration is for a 3-way link aggregation:
TABLE 12  Example - Configuration for a 3-way Link Aggregation

<table>
<thead>
<tr>
<th>Devices</th>
<th>Datalink</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>igb1, igb2, igb3</td>
<td>aggr1 (LACP aggregation)</td>
<td>phobos (192.168.2.81/22)</td>
</tr>
</tbody>
</table>

The datalink entity (which we named "aggr1") groups the network devices in a configurable way (LACP aggregation policy). The interface entity (which we named "phobos") provides configurable IP address settings, which it makes available on the network via the datalink. The network devices (named "igb1", "igb2", ..., by the system) have no direct settings. Datalinks are required to complete the network configuration, whether they apply specific settings to the network devices or not.

**Configuring Network Devices**

Network devices are created by the system to represent the available network or InfiniBand ports. They have no configuration settings of their own.

**Configuring Network Datalinks**

Network datalinks manage devices, and are used by interfaces. They support:

- **Link Aggregation Control Protocol (LACP)** - LACP is used to bundle multiple network devices such that they behave as one. This improves performance (by increasing bandwidth) and reliability (by protecting from network port failure); however, the appliance must be connected to a switch that supports LACP and has it enabled for those ports.

- **InfiniBand (IB) Partitions** - InfiniBand partitions connect to logically isolated IB fabric domains.

- **Virtual LANs (VLANs)** - VLANs are used to improve local network security and isolation. VLANs are recommended for administering the appliance; otherwise, use VNICS.

- **Virtual Network Interface Cards (VNICS)** - VNICS allow single or aggregated Ethernet datalinks to be split into multiple virtual (Ethernet) datalinks. VNICS can be optionally tagged with VLAN IDs, and can allow physical network port sharing in a cluster. Step-by-step instructions can be found in “Clustering Considerations for Networking” on page 154 below.
**Note** - VNIC-based and VLAN-based datalinks cannot share the same VLAN ID.

The IEEE802.3ad (link aggregation) standard does not explicitly support aggregations across multiple switches, but some vendors provide multi-switch support via proprietary extensions. If a switch configured with those extensions conforms to the IEEE standard and the extensions are transparent to the end-nodes, its use is supported with the appliance. If an issue is encountered, Oracle support may require it to be reproduced on a single-switch configuration.

The following datalink settings are available:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Use the defined custom name. For example: “internal”, “external”, “adminnet”, etc.</td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td>Use the defined speed. Valid values are auto, 10, 100, 1000 and 10000, representing autonegotiation, forced 10Mbit/sec, forced 100Mbit/sec, forced 1Gbit/sec and forced 10Gbit/sec. Speed and duplex must be either both forced to specific values or both set to autonegotiate. Not all networking devices support forcing to all possible speed/duplex combinations. Disabling autonegotiation is strongly discouraged. However, if the switch has autonegotiation disabled, it may be necessary to force speed (and duplex) to ensure the datalink runs at the expected speed and duplex.</td>
</tr>
<tr>
<td><strong>Duplex</strong></td>
<td>Use the defined transmission direction. Valid CLI values are auto, half, and full, representing autonegotiation, half- and full-duplex respectively. Speed and duplex must be either both forced to specific values or both set to autonegotiate.</td>
</tr>
<tr>
<td><strong>VLAN</strong></td>
<td>Use VLAN headers.</td>
</tr>
<tr>
<td><strong>VLAN ID</strong></td>
<td>Use the defined VLAN identifier; optional for VNICs.</td>
</tr>
<tr>
<td><strong>VNIC</strong></td>
<td>Use a VNIC.</td>
</tr>
<tr>
<td><strong>MTU</strong></td>
<td>Use the defined maximum transmission unit (MTU) size. The default MTU is 1500 bytes. Specify a lower MTU (minimum 1280) to leave packet headroom (for example, for tunneling protocols). Specify a larger MTU (maximum 9000) to improve network performance. All systems and switches on the same LAN must be configured with the chosen MTU. After the MTU value is set and the new network configuration is committed to the system, you can return to the network screen and view the datalink status to see the exact MTU value in bytes that was selected. Note that a VLAN or VNIC cannot be configured with an MTU value larger than that of the underlying datalink.</td>
</tr>
<tr>
<td><strong>LACP Aggregation</strong></td>
<td>Use multiple network device LACP aggregation.</td>
</tr>
<tr>
<td><strong>LACP Policy</strong></td>
<td>Use the defined LACP policy for selecting an outbound port. L2 hashes the source and destination MAC address;</td>
</tr>
</tbody>
</table>
### Network Configuration

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3 uses the source and destination IP address; L4 uses the source and destination transport level port</td>
<td></td>
</tr>
<tr>
<td>LACP Mode</td>
<td>Use the defined LACP communication mode. Active mode will send and receive LACP messages to negotiate connections and monitor the link status. Passive mode will listen for LACP messages only. Off mode will use the aggregated link but not detect link failure or switch configuration changes. Some network switch configurations, including Cisco Etherchannel, do not use the LACP protocol; the LACP mode should be set to “off” when using non-LACP aggregation in your network.</td>
</tr>
<tr>
<td>LACP Timer</td>
<td>Use the defined interval between LACP messages for Active mode.</td>
</tr>
<tr>
<td>IB Partition</td>
<td>Use IB Partitions.</td>
</tr>
<tr>
<td>Partition Key</td>
<td>Use the partition (fabric domain) in which the underlying port device is a member. The partition key (pkey) is found on and configured by the subnet manager. The pkey may be defined before configuring the subnet manager but the datalink will remain “down” until the subnet partition has been properly configured with the port GUID as a member. It is important to keep partition membership for HCA ports consistent with “Working with IPMP” on page 241 and “Configuring Clustering” on page 141 rules on the subnet manager.</td>
</tr>
<tr>
<td>IB Link Mode</td>
<td>Use the defined IB Link Mode. There are two modes: Unreliable Datagram and Connected. Unreliable Datagram lets a local queue pair communicate with multiple other queue pairs on any host and messages are communicated unacknowledged at the IB layer. Unreliable Datagram mode uses an MTU of 2044. Connected mode uses IB queue pairs and dedicates a local queue pair to communication with a dedicated remote queue pair. Connected mode uses an MTU of 65520 and can provides higher throughput than Unreliable Datagram.</td>
</tr>
</tbody>
</table>

### Configuring Network Interfaces

Network interfaces configure IP addresses via datalinks. The following are supported:

- IPv4 and IPv6 protocols.
- IPMP - IP MultiPathing, to improve network reliability by allowing IP addresses to automatically migrate from failed to working datalinks.

The following interface settings are available:
### TABLE 14  Interface Settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Custom name for the interface</td>
</tr>
<tr>
<td>Allow Administration</td>
<td>Allow connections to the appliance administration BUI or CLI over this interface. If your network environment included a separate administration network, this could be enabled for the administration network only to improve security.</td>
</tr>
<tr>
<td>Enable Interface</td>
<td>Enable this interface to be used for IP traffic. If an interface is disabled, the appliance will no longer send or receive IP traffic over it, or make use of any IP addresses configured on it. At present, disabling an active IP interface in an IPMP group will not trigger activation of a standby interface.</td>
</tr>
<tr>
<td>IPv4 Configure with</td>
<td>Either “Static Address List” manually entered, or “DHCP” for dynamically requested</td>
</tr>
<tr>
<td>IPv4 Address/Mask</td>
<td>One or more IPv4 addresses in CIDR notation (192.168.1.1/24)</td>
</tr>
<tr>
<td>IPv6 Configure with</td>
<td>Either “Static Address List” manually entered, or IPv6 AutoConfiguration to use automatically generated link-local address (and site-local if an IPv6 router responds)</td>
</tr>
<tr>
<td>IPv6 Address/Mask</td>
<td>One or more IPv6 addresses in CIDR notation (1080::8:800:200C:417A/32)</td>
</tr>
<tr>
<td>IP MultiPathing Group</td>
<td>Configure IP multipathing, where a pool of datalinks can be used for redundancy</td>
</tr>
</tbody>
</table>

### Configuring Network IP MultiPathing (IPMP)

IP MultiPathing groups are used to provide IP addresses that will remain available in the event of an IP interface failure (such as a physical wire disconnection or a failure of the connection between a network device and its switch) or in the event of a path failure between the system and its network gateways. The system detects failures by monitoring the IP interface's underlying datalink for link-up and link-down notifications, and optionally by probing using test addresses that can be assigned to each IP interface in the group, described below. Any number of IP interfaces can be placed into an IPMP group so long as they are all on the same link (LAN, IB partition, or VLAN), and any number of highly-available addresses can be assigned to an IPMP group.

Each IP interface in an IPMP group is designated either active or standby:

- **Active** - The IP interface will be used to send and receive data so long as IPMP has determined it is functioning correctly.
- **Standby** - The IP interface will only be used to send and receive data if an active interface (or a previously activated standby) stops functioning.

Multiple active and standby IP interfaces can be configured, but each IPMP group must be configured with at least one active IP interface. IPMP will strive to activate as many standbys...
as necessary to preserve the configured number of active interfaces. For example, if an IPMP group is configured with two active interfaces and two standby interfaces and all interfaces are functioning correctly, only the two active interfaces will be used to send and receive data. If an active interface fails, one of the standby interfaces will be activated. If the other active interface fails (or the activated standby fails), the second standby interface will be activated. If the active interfaces are subsequently repaired, the standby interfaces will again be deactivated.

IP interface failures can be discovered by either link-based detection or probe-based detection (i.e., a test address is configured).

If probe-based failure detection is enabled on an IP interface, the system will determine which target systems to probe dynamically. First, the routing table will be scanned for gateways (routers) on the same subnet as the IP interface's test address and up to five will be selected. If no gateways on the same subnet were found, the system will send a multicast ICMP probe (to 224.0.0.1 for IPv4 or ff02::1 for IPv6) and select the first five systems on the same subnet that respond. Therefore, for network failure detection and repair using IPMP, you should be sure that at least one neighbor on each link or the default gateway responds to ICMP echo requests.

IPMP works with both IPv4 and IPv6 address configurations. In the case of IPv6, the interface's link-local address is used as the test address.

Note - Do not use probe-based failure detection when there no systems (other than the cluster peer) on the same subnet as the IPMP test addresses that are configured to answer ICMP echo requests.

The system will probe selected target systems in round-robin fashion. If five consecutive probes are unanswered, the IP interface will be considered failed. Conversely, if ten consecutive probes are answered, the system will consider a previously failed IP interface as repaired. You can set the system's IPMP probe failure detection time from the IPMP screen. This time indirectly controls the probing rate and the repair interval -- for instance, a failure detection time of 10 seconds means that the system will send probes at roughly two second intervals and that the system will need 20 seconds to detect a probe-based interface repair. You cannot directly control the system's selected targeted systems, though it can be indirectly controlled through the routing table.

The system will monitor the routing table and automatically adjust its selected target systems as necessary. For instance, if the system using multicast-discovered targets but a route is subsequently added that has a gateway on the same subnet as the IP interface's test address, the system will automatically switch to probing the gateway. Similarly, if multicast-discovered targets are being probed, the system will periodically refresh its set of chosen targets (e.g., because some previously selected targets have become unresponsive).

For step-by-step instructions on building IPMP groups, see: “Working with IPMP” on page 241.

For information about private local interfaces, see “Configuring Clustering” on page 141.
Configuring Network Performance and Availability

IPMP and link aggregation are different technologies available in the appliance to achieve improved network performance as well as maintain network availability. In general, you deploy link aggregation to obtain better network performance, while you use IPMP to ensure high availability. The two technologies complement each other and can be deployed together to provide the combined benefits of network performance and availability.

In link aggregations, incoming traffic is spread over the multiple links that comprise the aggregation. Thus, networking performance is enhanced as more NICs are installed to add links to the aggregation. IPMP's traffic uses the IPMP interface's data addresses as they are bound to the available active interfaces. If, for example, all the data traffic is flowing between only two IP addresses but not necessarily over the same connection, then adding more NICs will not improve performance with IPMP because only two IP addresses remain usable.

Performance can be affected by the number of VNICs/VLANs configured on a datalink for a given device, as well as by using a VLAN ID. Configuring multiple VNICs over a given device may impact the performance of all datalinks over that device by up to five percent, even when VNICs are not in use. If more than eight VNICs/VLANs are configured over a given datalink, performance may degrade significantly. Also, if a datalink uses a VLAN ID, all datalink performance for that device may be impacted by an additional five percent.

Configuring Network Routing

The system provides a single IP routing table, consisting of a collection of routing table entries. When an IP packet needs to be sent to a given destination, the system selects the routing entry whose destination most closely matches the packet's destination address (subject to the system's multihoming policy; see below). It then uses the information in the routing entry to determine what IP interface to send the packet on and, if the destination is not directly reachable, the next-hop gateway to use. If no routing entries match the destination, the packet will be dropped. If multiple routing entries tie for closest match (and are not otherwise prioritized by multihoming policy), the system will load-spread across those entries on a per-connection basis.

The system does not act as a router.

The routing table is comprised of routing entries, each of which has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination</td>
<td>Range of IP destination addresses (in CIDR notation) that can match the route</td>
<td>192.168.0.0/22</td>
</tr>
<tr>
<td>Gateway</td>
<td>Next hop (IP address) to send the packet to (except for &quot;system&quot; routes -- see below)</td>
<td>192.168.2.80</td>
</tr>
</tbody>
</table>
A routing entry with a "destination" field of 0.0.0.0/0 matches any packet (if no other route matches more precisely), and is thus known as a 'default' route. In the BUI, default routes are distinguished from non-default routes by an additional property:

**TABLE 16**  Distinguishing Default from Non-default Routes

<table>
<thead>
<tr>
<th>Kind</th>
<th>Route kind</th>
<th>Default, Network</th>
</tr>
</thead>
</table>

As above, a given packet will be sent on the IP interface specified in the routing entry's "interface" field. If an IPMP interface is specified, then one of the active IP interfaces in the IPMP group will be chosen randomly on a per-connection basis and automatically refreshed if the chosen IP interface subsequently becomes unusable. Conversely, if a given IP interface is part of an IPMP group, it cannot be specified in the "interface" field because such a route would not be highly-available.

Routing entries come from a number of different origins, as identified by the "type" field. Although the origin of a routing entry has no bearing on how it is used by the system, its origin does control if and how it can be edited or deleted. The system supports the following types of routes:

**TABLE 17**  Supported Route Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>Created and managed by the appliance administrator.</td>
</tr>
<tr>
<td>System</td>
<td>Created automatically by the appliance as part of enabling an IP interface.</td>
</tr>
<tr>
<td></td>
<td>A system route will be created for each IP subnet the appliance can directly reach.</td>
</tr>
<tr>
<td></td>
<td>Since these routes are directly reachable, the &quot;gateway&quot; field instead identifies the appliance's IP address on that subnet.</td>
</tr>
<tr>
<td>DHCP</td>
<td>Created automatically by the appliance part of enabling an IP interface that is configured to use DHCP. A DHCP route will be created for each default route provided by the DHCP server.</td>
</tr>
<tr>
<td>Dynamic</td>
<td>Created automatically by the appliance via the RIP and RIPng dynamic routing protocols (if enabled).</td>
</tr>
</tbody>
</table>

One additional type identifies a static route that cannot currently be used:
### Table 18: Unavailable Static Route Type

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inactive</td>
<td>Previously created static route associated with a disabled or offline IP interface.</td>
</tr>
</tbody>
</table>

### Table 19: Routing Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multihoming model</td>
<td>Controls the system policy for accepting and transmitting IP packets when multiple IP interfaces are simultaneously enabled. Allowed values are &quot;loose&quot; (default), &quot;adaptive&quot;, and &quot;strict&quot;. See the discussion below.</td>
</tr>
</tbody>
</table>

If a system is configured with more than one IP interface, then there may be multiple equivalent routes to a given destination, forcing the system to choose which IP interface to send a packet on. Similarly, a packet may arrive on one IP interface, but be destined to an IP address that is hosted on another IP interface. The system's behavior in such situations is determined by the selected multihoming policy. Three policies are supported:

### Table 20: Multihoming Policies

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loose</td>
<td>Do not enforce any binding between an IP packet and the IP interface used to send or receive it: 1) An IP packet will be accepted on an IP interface so long as its destination IP address is up on the appliance. 2) An IP packet will be transmitted over the IP interface tied to the route that most specifically matches an IP packet's destination address, without any regard for the IP addresses hosted on that IP interface. If no eligible routes exist, drop the packet.</td>
</tr>
<tr>
<td>Adaptive</td>
<td>Identical to loose, except prefer routes with a gateway address on the same subnet as the packet's source IP address: 1) An IP packet will be accepted on an IP interface so long as its destination IP address is up on the appliance. 2) An IP packet will be transmitted over the IP interface tied to the route that most specifically matches an IP packet's destination address. If multiple routes are equally specific, prefer routes that have a gateway address on the same subnet as the packet's source address. If no eligible routes exist, drop the packet.</td>
</tr>
<tr>
<td>Strict</td>
<td>Require a strict binding between an IP packet and the IP interface used to send or receive it: 1) An IP packet will be accepted on an IP interface so long as its destination IP address is up on that IP interface. 2) An IP packet will only be transmitted over an IP interface if its source IP address is up on that IP interface. To enforce this, when matching against the available routes, the appliance...</td>
</tr>
</tbody>
</table>
When selecting the multihoming policy, a key consideration is whether any of the appliance's IP interfaces will be dedicated to administration (for example, for dedicated BUI access) and thus accessed over a separate administration network. In particular, if a default route is created to provide remote access to the administration network, and a separate default route is created to provide remote access to storage protocols, then the default system policy of "loose" may cause the administrative default route to be used for storage traffic. By switching the policy to "adaptive" or "strict", the appliance will consider the IP address associated with the request as part of selecting the route for the reply. If no route can be found on the same IP interface, the "adaptive" policy will cause the system to use any available route, whereas the "strict" policy will cause the system to drop the packet.

**Network Configuration Using the BUI**

When using the BUI to reconfigure networking, the system makes every effort to preserve the current networking connection to your browser. However, some network configuration changes such as deleting the specific address to which your browser is connected, will unavoidably cause the browser to lose its connection. For this reason it is recommended that you assign a particular IP address and network device for use by administrators and always leave the address configured. You can also perform particularly complex network reconfiguration tasks from the CLI over the serial console if necessary.

The following icons are used in the Configuration>Network section:

<table>
<thead>
<tr>
<th>icon</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Add new datalink/interface/route" /></td>
<td>Add new datalink/interface/route</td>
</tr>
<tr>
<td><img src="image" alt="Edit datalink/interface/route settings" /></td>
<td>Edit datalink/interface/route settings</td>
</tr>
<tr>
<td><img src="image" alt="Editing disabled" /></td>
<td>Editing disabled</td>
</tr>
<tr>
<td><img src="image" alt="Destroy datalink/interface/route" /></td>
<td>Destroy datalink/interface/route</td>
</tr>
<tr>
<td><img src="image" alt="Destruction disabled" /></td>
<td>Destruction disabled</td>
</tr>
<tr>
<td><img src="image" alt="Drag-and-drop icon" /></td>
<td>Drag-and-drop icon</td>
</tr>
<tr>
<td><img src="image" alt="Connected network port" /></td>
<td>connected network port</td>
</tr>
<tr>
<td><img src="image" alt="Connected network port with I/O activity" /></td>
<td>connected network port with I/O activity</td>
</tr>
</tbody>
</table>
At top right is local navigation for Configuration, Addresses and Routing, which display alternate configuration views.

The Configuration page is shown by default, and lists Devices, Datalinks and Interfaces, along with buttons for administration. Mouse-over an entry to expose an additional icon, and click on any entry to highlight other components that are associated with it.

The Devices list shows links status on the right, as well as an icon to reflect the state of the network port. If ports appear disconnected, check that they are plugged into the network properly.

To configure an IP address on a network devices, first create a datalink, and then create an interface to use that datalink. The icon may be used to do both, which will display dialogs for the Datalink and Interface properties.

There is more than one way to configure a network interface. Try clicking on the icon for a device, then dragging it to the datalink table. Then drag the datalink over to the interfaces table.

<table>
<thead>
<tr>
<th>icon</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Disconnected network port" /></td>
<td>disconnected network port (link down, cable problem?)</td>
</tr>
<tr>
<td><img src="image" alt="Active InfiniBand port" /></td>
<td>active InfiniBand port</td>
</tr>
<tr>
<td><img src="image" alt="Active InfiniBand port with I/O activity" /></td>
<td>active InfiniBand port with I/O activity</td>
</tr>
<tr>
<td><img src="image" alt="Inactive InfiniBand port (down, init, or arm state)" /></td>
<td>inactive InfiniBand port (down, init, or arm state)</td>
</tr>
<tr>
<td><img src="image" alt="InfiniBand partition device is up" /></td>
<td>InfiniBand partition device is up</td>
</tr>
<tr>
<td><img src="image" alt="InfiniBand partition device is down (subnet manager problem)" /></td>
<td>InfiniBand partition device is down (subnet manager problem)</td>
</tr>
<tr>
<td><img src="image" alt="Network datalink" /></td>
<td>network datalink</td>
</tr>
<tr>
<td><img src="image" alt="Network datalink VLAN or VNIC" /></td>
<td>network datalink VLAN or VNIC</td>
</tr>
<tr>
<td><img src="image" alt="Network datalink aggregation" /></td>
<td>network datalink aggregation</td>
</tr>
<tr>
<td><img src="image" alt="Network datalink aggregation VLAN or VNIC" /></td>
<td>network datalink aggregation VLAN or VNIC</td>
</tr>
<tr>
<td><img src="image" alt="Network datalink IB partition" /></td>
<td>network datalink IB partition</td>
</tr>
<tr>
<td><img src="image" alt="Interface is being used to send and receive packets (either up or degraded)" /></td>
<td>interface is being used to send and receive packets (either up or degraded)</td>
</tr>
<tr>
<td><img src="image" alt="Interface has been disabled by the user" /></td>
<td>interface has been disabled by the user</td>
</tr>
<tr>
<td><img src="image" alt="Interface is offline (owned by the cluster peer)" /></td>
<td>interface is offline (owned by the cluster peer)</td>
</tr>
<tr>
<td><img src="image" alt="Interface has failed or has been configured with a duplicate IP address" /></td>
<td>interface has failed or has been configured with a duplicate IP address</td>
</tr>
</tbody>
</table>
Other moves are possible. This can be helpful for complex configurations, where valid moves are highlighted.

This page shows a summary table of the current network configuration, with fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Datalink</td>
<td>Datalink name and detail summary</td>
<td>datalink1 (via igb0)</td>
</tr>
<tr>
<td>Network Interface</td>
<td>Interface name and details summary</td>
<td>IPv4 DHCP, via datalink1</td>
</tr>
<tr>
<td>Network Addresses</td>
<td>Addresses hosted by this interface</td>
<td>192.168.2.80/22</td>
</tr>
<tr>
<td>Host Names</td>
<td>Resolved host names for the network addresses</td>
<td>caji.sf.example.com</td>
</tr>
</tbody>
</table>

This page provides configuration of the IP routing table and associated properties, as discussed above. By default, all entries in the routing table are shown, but the table can be filtered by type by using the subnavigation bar.

To check a specific route, in the CLI use `traceroute`.

```
zfssa-source:> traceroute 10.80.198.102
traceroute: Warning: Multiple interfaces found; using 10.80.198.101 @ igb3
traceroute to 10.80.198.102 (10.80.198.102), 30 hops max, 40 byte packets
1 10.80.198.1 (10.80.198.1) 6.490 ms 0.924 ms 0.834 ms
2 10.80.198.102 (10.80.198.102) 0.152 ms 0.118 ms 0.099 ms
zfssa-target:> traceroute 10.80.198.101
traceroute: Warning: Multiple interfaces found; using 10.80.198.102 @ igb3
traceroute to 10.80.198.101 (10.80.198.101), 30 hops max, 40 byte packets
1 10.80.198.1 (10.80.198.1) 1.031 ms 0.905 ms 0.769 ms
2 10.80.198.101 (10.80.198.101) 0.158 ms 0.111 ms 0.109 ms
```

▼ Creating a Single Port Interface Using the BUI

1. Click the Datalinks + icon.
2. Optionally set name and select custom MTU radio button (typing 9000 in the text box).
3. Choose a device from the Devices list.
4. Click "APPLY". The datalink will appear in the Datalinks list.
5. Click the Interface + icon.
6. Set desired properties, and choose the datalink previously created.
7. Click "APPLY". The interface will appear in the Interfaces list.

8. The running appliance network configuration has not yet changed. When you are finished configuring interfaces, click "APPLY" at the top to commit the configuration.

▼ Modifying an Interface Using the BUI

1. Click the edit icon on either the datalink or the interface.
2. Change settings to desired values.
3. Click "APPLY" on the dialog.
4. Click "APPLY" at the top of the page to commit the configuration.

▼ Creating a Single Port Interface Using the BUI

1. Mouse over a device and click the drag-and-drop icon (_drag-and-drop_icon_).
2. Drag it to the Datalink list and release.
3. Optionally set name and jumbo MTU.
4. Click "APPLY".
5. Now Drag the datalink over to the Interfaces list.
6. Set desired properties, and click "APPLY".
7. Click "APPLY" at the top of the screen to commit the configuration.

▼ Creating an LACP Aggregated Link Interface Using the BUI

1. Click the Datalinks icon.
2. Optionally set the datalink name.
3. Select LACP Aggregation.
4. Select two or more devices from the Devices list, and click "APPLY".
Creating an IPMP Group Using Probe-Based and Link-State Failure Detection

Do not use probe-based failure detection when there no systems (other than the cluster peer) on the same subnet as the IPMP test addresses that are configured to answer ICMP echo requests.

1. Create one or more "underlying" IP interfaces that will be used as components of the IPMP group. Each interface must have an IP address to be used as the probe source (see separate task to create a single-port interfaces above).

2. Click the Interface icon.

3. Optionally change the name of the interface.

4. Click the IP MultiPathing Group check box.

5. Click the Use IPv4 Protocol or/and the Use IPv6 Protocol and specify the IP addresses for the IPMP interface.

6. Choose the interfaces created in the first step from the Interfaces list.

7. Set each chosen interface to be either "Active" or "Standby", as desired.

8. Click "APPLY".

Creating an IPMP Group Using Link-State Only Failure Detection

1. Create one or more "underlying" IP interfaces with the IP address 0.0.0.0/8 to be used as the components of the IPMP group (see separate task to create a single-port interfaces above).

2. Click the Interface icon.

3. Optionally change the name of the interface.
4. Click the IP MultiPathing Group check box.

5. Click the Use IPv4 Protocol or/and the Use IPv6 Protocol and specify the IP addresses for the IPMP interface.

6. Choose the interfaces created in the first step from the Interfaces list.

7. Set each chosen interface to be either "Active" or "Standby", as desired.

8. Click "APPLY".

▼ Extending an LACP Aggregation Using the BUI

1. Mouse-over a device in the Devices list.

2. Click the icon, and drag the device onto an aggregation datalink, and release.

3. Click "APPLY" at the top of the page to commit this configuration.

▼ Extending an IPMP group Using the BUI

1. Mouse-over an interface in the Interfaces list.

2. Click the icon, and drag the device onto an IPMP interface, and release.

3. Click "APPLY" at the top of the page to commit this configuration.

▼ Creating an InfiniBand Partition Datalink and Interface Using the BUI

1. Click the Datalink icon.

2. Optionally, set name.

3. Click the IB Partition checkbox.

4. Choose a device from the Partition Devices list.

5. Enter a four-digit hexadecimal number for the partition key, which must match what was configured on the InfiniBand subnet manager.

6. Choose link mode from the drop down menu.
Creating a VNIC Without a VLAN ID for Clustered Controllers Using the BUI

7. Click "APPLY". The new partition datalink will appear in the Datalinks list.

8. Click the Interface icon.

9. Set desired properties, and choose the datalink previously created.

10. Click "APPLY". The interface will appear in the Interfaces list.

11. The running appliance network configuration has not yet changed. When you are finished configuring interfaces, click "APPLY" at the top to commit the configuration.

Creating a VNIC Without a VLAN ID for Clustered Controllers Using the BUI

This example is for an active-active configuration with half of the network ports on standby. This task creates an IP interface over a device datalink and assigns it to a head. A VNIC is built on top of the same datalink, and an IP interface is configured on top of the VNIC and assigned to the other head. Configuring one instead of multiple VNICs over a given datalink ensures peak performance. Traffic flows over the cable associated with the underlying active port on one head, as well as the underlying standby port on the other head. Thus, the otherwise idle standby port can be used with VNICs.

1. When the cluster is in state AKCS_CLUSTERED, click the Datalinks icon.

2. Optionally, set name and MTU.

3. Choose a device from the Devices list and click "APPLY". The datalink appears in the Datalinks list.

4. Click the Interface icon.

5. Set desired properties, choose the datalink previously created, and click "APPLY". The interface appears in the Interfaces list.

6. Click the Datalinks icon.

7. Select the VNIC checkbox, optionally set name and MTU (equal to or less than the value in step 2), and click "APPLY". The new VNIC datalink appears in the Datalinks list.

8. Click the Interface icon.
9. Set desired properties, choose the VNIC datalink previously created, and click "APPLY". The interface appears in the Interfaces list.

10. The running appliance network configuration has not yet changed. When you are finished configuring interfaces, click "APPLY" at the top to commit the configuration.

11. Click the Cluster tab. The two newly created interfaces appear in the Resource section with default owners.

12. Use the Owner pull-down list to assign one of the two interfaces to the other head and click "APPLY".

Creating VNICs with the Same VLAN ID for Clustered Controllers Using the BUI

This example is for an active-active configuration with half of the network ports on standby. This task creates two VNICs with identical VLAN IDs on top of the same device datalink. Each VNIC is configured with an interface, and each interface is assigned to a different head. Traffic flows over the cable associated with the underlying active port on one head, as well as the underlying standby port on the other head. Thus, the otherwise idle standby port can be used with VNICs.

1. When the cluster is in state AKCS_CLUSTERED, click the Datalinks icon.

2. Select the VNIC checkbox, optionally set name and MTU, set the VLAN ID, choose a device from the Devices list, and click "APPLY". The new VNIC datalink appears in the Datalinks list.

3. Click the Interface icon.

4. Set desired properties, choose the VNIC datalink previously created, and click "APPLY". The interface appears in the Interfaces list.

5. Create another VNIC as described in steps 1 and 2 with the same Device and VLAN ID, and create an interface for it as described in steps 3 and 4.

6. The running appliance network configuration has not yet changed. When you are finished configuring interfaces, click "APPLY" at the top to commit the configuration.

7. Click the Cluster tab. The two newly created interfaces appear in the Resource section with default owners.
8. Use the Owner pull-down list to assign one of the two interfaces to the other head and click "APPLY".

▼ Adding a Static Route Using the BUI

1. Go to Configuration>Network>Routing.
2. Click the add icon.
3. Fill in the properties as described earlier.
4. Click "ADD". The new route will appear in the table.

▼ Deleting a Static Route Using the BUI

1. Go to Configuration>Network>Routing.
2. Mouse-over the route entry, then click the trash icon on the right.

Network Configuration Using the CLI

Network configuration is under the configuration net, which has sub commands for devices, datalinks, interfaces, and routing. The show command can be used with each to show the current configuration:

caji:> configuration net
caji:configuration net> devices show
Devices:

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>UP</th>
<th>SPEED</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>igb0</td>
<td>true</td>
<td>1000 Mbit/s</td>
<td>0:14:4f:9a:b9:0</td>
</tr>
<tr>
<td>igb1</td>
<td>true</td>
<td>1000 Mbit/s</td>
<td>0:14:4f:9a:b9:1</td>
</tr>
<tr>
<td>igb2</td>
<td>true</td>
<td>1000 Mbit/s</td>
<td>0:14:4f:9a:b8:fe</td>
</tr>
<tr>
<td>igb3</td>
<td>true</td>
<td>1000 Mbit/s</td>
<td>0:14:4f:9a:b8:ff</td>
</tr>
</tbody>
</table>

caji:configuration net> datalinks show
Datalinks:

<table>
<thead>
<tr>
<th>DATALINK CLASS</th>
<th>LINKS</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>igb0 device</td>
<td>igb0</td>
<td>datalink1</td>
</tr>
</tbody>
</table>

caji:configuration net> interfaces show
Interfaces:

<table>
<thead>
<tr>
<th>INTERFACE</th>
<th>STATE</th>
<th>CLASS</th>
<th>LINKS</th>
<th>ADDRS</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>igb0</td>
<td>up</td>
<td>ip</td>
<td>igb0</td>
<td>192.168.2.80/22</td>
<td>caji</td>
</tr>
</tbody>
</table>
Deleting a Static Route Using the BUI

caji:configuration net> routing show
Properties:
  multihoming = loose

Routes:

<table>
<thead>
<tr>
<th>ROUTE</th>
<th>DESTINATION</th>
<th>GATEWAY</th>
<th>INTERFACE</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>route-000</td>
<td>0.0.0.0/0</td>
<td>192.168.1.1</td>
<td>igb0</td>
<td>dhcp</td>
</tr>
<tr>
<td>route-001</td>
<td>192.168.0.0/22</td>
<td>192.168.2.142</td>
<td>igb0</td>
<td>system</td>
</tr>
</tbody>
</table>

Type help in each section to see the relevant commands for creating and configuring datalinks, interfaces, and routes. Subcommands that are valid in this context:

- **help [topic]** => Get context-sensitive help. If [topic] is specified, it must be one of "builtins", "commands","general", "help", "script" or "properties".
- **show** => Show information pertinent to the current context
- **commit** => Commit current state, including any changes
- **abort** => Abort creation of "vnic"
- **done** => Finish operating on "vnic"
- **get [prop]** => Get value for property [prop]. ("help properties" for valid properties.) If [prop] is not specified, returns values for all properties.
- **set [prop]** => Set property [prop] to [value]. ("help properties" for valid properties.) For properties taking list values, [value] should be a comma-separated list of values.
- **available** => Get values that can be assigned to the links parameter when creating a network component.

The available command is used to see what values can be assigned to the links parameter when creating a network component. The following shows the output from the CLI command available:

caji:configuration net datalinks> device
caji:configuration net datalinks device (uncommitted)> available
  igb7,igb6

caji:configuration net datalinks> vnic
caji:configuration net datalinks vnic (uncommitted)> available
  igb5,igb4,aggr2,aggr1

caji:configuration net datalinks> vlan
caji:configuration net datalinks vlan (uncommitted)> available
  igb5,igb4,aggr2,aggr1
Deleting a Static Route Using the BUI

The following demonstrates creating a datalink using the device command, and interface using the ip command:

```
caji:configuration net> datalinks
caji:configuration net datalinks> device
caji:configuration net datalinks device (uncommitted)> set links=igb1
   links = igb1 (uncommitted)
caji:configuration net datalinks device (uncommitted)> set label=datalink2
   label = datalink2 (uncommitted)
caji:configuration net datalinks device (uncommitted)> set mtu=9000
   mtu = 9000 (uncommitted)
caji:configuration net datalinks device (uncommitted)> commit

Datalinks:
     DATALINK CLASS          LINKS       LABEL
      igb0 device         igb0        datalink1
      igb1 device         igb1        datalink2
```

caji:configuration net datalinks> cd ..
caji:configuration net> interfaces
caji:configuration net interfaces> ip
caji:configuration net interfaces ip (uncommitted)> set label="caji2"
   label = caji2 (uncommitted)
caji:configuration net interfaces ip (uncommitted)> set links=igb1
   links = igb1 (uncommitted)
caji:configuration net interfaces ip (uncommitted)> set v4addrs=10.0.1.1/8
   v4addrs = 10.0.1.1/8 (uncommitted)
caji:configuration net interfaces ip (uncommitted)> commit

Interfaces:
    INTERFACE STATE  CLASS LINKS    ADDR         LABEL
      igb0 up     ip    igb0  192.168.2.80/22  caji
      igb1 up     ip    igb1  10.0.1.1/8      caji2
```

The following demonstrates creating a default route via 10.0.1.2 over the new igb1 IP interface:

```
caji:configuration net routing> create
caji:configuration net route (uncommitted)> set family=IPv4
   family = IPv4 (uncommitted)
```
Adding a Static Route Using the CLI

1. Go to configuration net routing.
2. Enter create.
3. Type show to list required properties, and set each.
4. Enter commit.

Deleting a Static Route Using the CLI

1. Go to configuration net routing.
2. Type show to list routes, and route names (e.g., route-002).
3. Enter destroy route name.

Changing the Multihoming Property to Strict Using the CLI

1. Go to configuration net routing.
2. Enter set multihoming=strict.
3. Enter commit.

Storage Configuration

Storage is configured in pools that are characterized by their underlying data redundancy, and provide space that is shared across all filesystems and LUNs. More information about how storage pools relate to individual filesystems or LUNs can be found in “Shares”.

```bash
caji:configuration net route (uncommitted)> set destination=0.0.0.0
destination = 0.0.0.0 (uncommitted)
caji:configuration net route (uncommitted)> set mask=0
mask = 0 (uncommitted)
caji:configuration net route (uncommitted)> set interface=igb1
interface = igb1 (uncommitted)
caji:configuration net route (uncommitted)> set gateway=10.0.1.2
gateway = 10.0.1.2 (uncommitted)
caji:configuration net route (uncommitted)> commit
```
Each node can have any number of pools, and each pool can be assigned ownership independently in a cluster. While arbitrary number of pools are supported, creating multiple pools with the same redundancy characteristics owned by the same cluster head is not advised. Doing so will result in poor performance, suboptimal allocation of resources, artificial partitioning of storage, and additional administrative complexity. Configuring multiple pools on the same host is only recommended when drastically different redundancy or performance characteristics are desired, for example a mirrored pool and a RAID-Z pool. With the ability to control access to log and cache devices on a per-share basis, the recommended mode of operation is a single pool.

Pools can be created by configuring a new pool, or importing an existing pool. Importing an existing pool is only used to import pools previously configured on an Oracle ZFS Storage Appliance, and is useful in case of accidental reconfiguration, moving of pools between head nodes, or due to catastrophic head failure.

When allocating raw storage to pools, keep in mind that filling pools completely will result in significantly reduced performance, especially when writing to shares or LUNs. These effects typically become noticeable once the pool exceeds 80% full, and can be significant when the pool exceeds 90% full. Therefore, best results will be obtained by over provisioning by approximately 20%. The Shares UI can be used to determine how much space is currently being used.

Drives within all of the chassis can be allocated individually; however, care should be taken when allocating disks from disk shelves to ensure optimal pool configurations. In general, fewer pools with more disks per pool are preferred because they simplify management and provide a higher percentage of overall usable capacity.

While the system can allocate storage in any increment desired, it is recommended that each allocation include a minimum of 8 disks across all disk shelves and ideally many more.

For optimal performance, use the following rules:

- **Rule 1** - All "data" disks contained within a head node or disk shelf must have the same rotational speed (media rotation rate). The appliance software will detect misconfigurations and generate a fault for the condition.

- **Recommendation 1** - Due to unpredictable performance issues, avoid mixing different disk rotational speeds within the same pool.

- **Recommendation 2** - For optimal performance, do not combine disk shelves with different disk rotational speeds on the same SAS fabric (HBA connection). Such a mixture operates correctly, but likely results in slower performance of the faster devices.

- **Recommendation 3** - When configuring storage pools that contain data disks of different capacities, ZFS will in some cases use the size of the smallest capacity disk for some or all of the disks within the storage pool, thereby reducing the overall expected capacity. The sizes used will depend on the storage profile, layout, and combination of devices. Avoid mixing different disk capacities within the same pool.

To configure storage, use the following sections:
Choose a Storage Profile

This action configures the storage pool. In the BUI, this is done by clicking the button next to the list of pools, at which point you are prompted for the name of the new pool. In the CLI, this is done by the `config` command, which takes the name of the pool as an argument.

**FIGURE 3**  Choose Storage Profile
After the task is started, storage configuration falls into two different phases: verification and configuration.

**FIGURE 4 Verify and Allocate Devices**

Verification ensures that all storage is attached and functioning. All storage devices must be connected and functioning before you can allocate them. If you allocate a pool with missing or failed devices, you will not be able to add the missing or failed devices later.

In a system without attached storage, all available drives are allocated by default. In an expandable system, disk shelves are displayed in a list along with the head node, and allocation can be controlled within each disk shelf. This may operate differently depending on the model of the head node or disk shelf.

You can select the following:

- **Device size** - Filters Data devices by logical size. By default, Any displays all available data devices.
- **Data devices** - Displays all available data devices, or the available number of the selected device size.

The number of disks allocated by default depends on the following:

- **The maximum number available** - When the attached storage only contains devices with the same size and rotational speed, or when one size is selected among multiple sizes
- **None** - When the attached storage contains a mixture of rotational speeds.

**Note** - It is strongly recommended that pools include only devices of the same size and rotational speed to provide consistent performance characteristics.

Drives within all of the chassis can be allocated individually; however, care should be taken when allocating disks from disk shelves to ensure optimal pool configurations. In general, fewer pools with more disks per pool are preferred because they simplify management and provide a higher percentage of overall usable capacity.
While the system can allocate storage in any increment desired, it is recommended that each allocation include a minimum of 8 disks across all disk shelves and ideally many more.

**Data Profile Configuration**

Once verification is completed, the next step involves choosing a storage profile that reflects the RAS and performance goals of your setup. The set of possible profiles presented depends on your available storage. The following table lists all possible profiles and their description.

<table>
<thead>
<tr>
<th>Data Profile</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dual Parity Options</strong></td>
<td></td>
</tr>
<tr>
<td>Triple mirrored</td>
<td>Data is triply mirrored, yielding a very highly reliable and high-performing system (for example, storage for a critical database). This configuration is intended for situations in which maximum performance and availability are required. Compared with a two-way mirror, a three-way mirror adds additional IOPs per stored block and higher level protection against failures. Note: A controller without expansion storage should not be configured with triple mirroring.</td>
</tr>
<tr>
<td>Double parity RAID</td>
<td>RAID in which each stripe contains two parity disks. As with triple mirroring, this yields high availability, as data remains available with the failure of any two disks. Double parity RAID is a higher capacity option than the mirroring options and is intended either for high-throughput sequential-access workloads (such as backup) or for storing large amounts of data with low random-read component.</td>
</tr>
<tr>
<td><strong>Single Parity Options</strong></td>
<td></td>
</tr>
<tr>
<td>Mirrored</td>
<td>Data is mirrored, reducing capacity by half, but yielding a highly reliable and high-performing system. Recommended when space is considered ample, but performance is at a premium (for example, database storage).</td>
</tr>
<tr>
<td>Single parity RAID, narrow stripes</td>
<td>RAID in which each stripe is kept to three data disks and a single parity disk. For situations in which single parity protection is acceptable, single parity RAID offers a much higher capacity option than simple mirroring. This higher capacity needs to be balanced against a lower random read capability than mirrored options. Single parity RAID can be considered for non-critical applications with a moderate random read component. For pure streaming workloads, give preference to the Double parity RAID option which has higher capacity and more throughput.</td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td></td>
</tr>
<tr>
<td>Striped</td>
<td>Data is striped across disks, with no redundancy. While this maximizes both performance and capacity, a single</td>
</tr>
</tbody>
</table>
Changing the Multihoming Property to Strict Using the CLI

<table>
<thead>
<tr>
<th>Data Profile</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data profile</td>
<td>description</td>
</tr>
<tr>
<td>disk failure will result in data loss.</td>
<td>This configuration is not recommended. For pure streaming workloads, consider using Double parity RAID.</td>
</tr>
<tr>
<td>Triple parity RAID, wide stripes</td>
<td>RAID in which each stripe has three disks for parity. This is the highest capacity option apart from Striped Data. Resilvering data after one or more drive failures can take significantly longer due to the wide stripes and low random I/O performance. As with other RAID configurations, the presence of cache can mitigate the effects on read performance. This configuration is not generally recommended.</td>
</tr>
</tbody>
</table>

For expandable systems, some profiles may be available with an 'NSPF' option. This stands for 'no single point of failure' and indicates that data is arranged in mirrors or RAID stripes such that a pathological disk shelf failure will not result in data loss. Note that systems are already configured with redundancy across nearly all components. Each disk shelf has redundant paths, redundant controllers, and redundant power supplies and fans. The only failure that NSPF protects against is disk backplane failure (a mostly passive component), or gross administrative misconduct (detaching both paths to one disk shelf). In general, adopting NSPF will result in lower capacity, as it has more stringent requirements on stripe width.

Log devices can be configured using only striped or mirrored profiles. Since log devices are only used in the event of node failure for data to be lost with unmirrored logs, it is necessary for both the device to fail and the node to reboot immediately after. This a highly-unlikely event, however mirroring log devices can make this effectively impossible, requiring two simultaneous device failures and node failure within a very small time window.

Note - When different sized log devices are in different chassis, only striped log profiles can be created.

Hot spares are allocated as a percentage of total pool size and are independent of the profile chosen (with the exception of striped, which doesn't support hot spares). Because hot spares are allocated for each storage configuration step, it is much more efficient to configure storage as a whole than it is to add storage in small increments.

In a cluster, cache devices are available only to the node which has the storage pool imported. In a cluster, it is possible to configure cache devices on both nodes to be part of the same pool. To do this, takeover the pool on the passive node, and then add storage and select the cache devices. This has the effect of having half the global cache devices configured at any one time. While the data on the cache devices will be lost on failover, the new cache devices can be used on the new node.

Note - Earlier software versions supported double parity with wide stripes. This has been supplanted by triple parity with wide stripes, as it adds significantly better reliability. Pools configured as double parity with wide stripes under a previous software version continue to be supported, but newly-configured or reconfigured pools cannot select that option.
Importing Existing Storage Pools

The import action allows you to import an unconfigured storage pool. A pool can be unconfigured because of an inadvertent action, factory reset, or service operation to recover user data. Importing a pool requires iterating over all attached storage devices and discovering any existing state. This can take a significant amount of time, during which no other storage configuration activities can take place. To import a pool in the BUI, click the IMPORT button in the storage configuration screen. To import a pool in the CLI, use the command `import` in the `configuration storage` context.

Once the discovery phase has completed, you will be presented with a list of available pools, including some identifying characteristics. If the storage has been destroyed or is incomplete, the pool will not be importable. Unlike storage configuration, the pool name is not specified at the beginning, but rather when selecting the pool. By default, the previous pool name is used, but you can change the pool name, either by clicking the name in the BUI or setting the 'name' property in the CLI.

Adding Storage

Use this action to add additional storage to your existing pool. The verification step is identical to the verification step during initial configuration. The storage must be added using the same profile that was used to configure the pool initially. If there is insufficient storage to configure the system with the current profile, some attributes can be sacrificed. For example, adding a single disk shelf to a double parity RAID-Z NSPF config makes it impossible to preserve NSPF characteristics. However, you can still add the disk shelf and create RAID stripes within the disk shelf, sacrificing NSPF in the process. To add storage to a pool in the BUI, click the ADD button in the storage configuration screen. To add storage to a pool in the CLI, use the command `add` in the `configuration storage` context. If there are multiple pools, first specify the pool using the command `set pool=pool name`.

Unconfiguring Storage

This removes any active filesystems and LUNs and unconfigure the storage pool, making the raw storage available for future storage configuration. This process can be undone by importing the unconfigured storage pool, provided the raw storage has not since been used as part of an active storage pool. To unconfigure the storage pool in the BUI, click the UNCONFIG button in the storage configuration screen. To unconfigure the storage pool in the CLI, use the command `unconfig` in the `configuration storage` context. If there are multiple pools, first specify the pool using the command `set pool=pool name`. 
Renaming a Storage Pool

To rename a storage pool, unconfigure it and then immediately import it with a new name. To import a pool in the BUI, click the IMPORT button in the storage configuration screen. To import a pool in the CLI, use the command `import` in the configuration storage context. By default, the previous pool name is used, but you can change the pool name, either by clicking the name in the BUI or setting the 'name' property in the CLI.

Storage Pool Scrub

The scrub action initiates the storage pool scrub process, which verifies all content to check for errors. If any unrecoverable errors are found, either through a scrub or through normal operation, the BUI displays the affected files. The scrub can also be stopped if necessary. To start the scrub process in the BUI, click the SCRUB button in the storage configuration screen. To start the scrub process in the CLI, use the command `configuration storage scrub start`. To stop the scrub process in the CLI, use the command `configuration storage scrub stop`.

 النبي Configuring a Storage Pool using the BUI

There are two ways to arrive at this task: either during initial configuration of the appliance, or at the Configuration->Storage screen.

1. Click the button above the list of storage pools
2. Enter a name for the storage pool.
3. At the "Allocate and verify storage" screen, configure the disk shelf allocation for the storage pool. If no disk shelves are detected, check your disk shelf cabling and power.
4. Click "COMMIT".
5. On the "Configure Added Storage" screen, select the desired data profile. Each is rated in terms of availability, performance and capacity, to help find the best configuration for your business needs.
6. Click "COMMIT".
Adding Cache Devices to an Existing Pool using the BUI

1. Install the new Readzilla or Logzilla device into the first available slot. See the “Oracle ZFS Storage Appliance Installation Guide” for slot locations.

2. In the BUI, go to Configuration > Storage.

3. From the Available Pools list, select the pool you're adding the device to. Be sure the pool is online.

4. Click the Add button to add the device to the pool.

5. Select the device you're adding to the pool, and click Commit.

6. Select the log profile (if applicable), and click Commit.

Adding Cache Devices to an Existing Pool using the CLI

1. Install the new Readzilla or Logzilla device into the first available slot. See the “Oracle ZFS Storage Appliance Installation Guide” for slot locations.

2. At the command line, enter:
   
   : poc:> configuration storage

3. Specify the pool you want to add the device to:
   
   : poc:configuration storage (pool_2)> set pool=pool_2
   : pool = pool_2
   : poc:configuration storage (pool_2)> add

4. A message reminds you to verify that the device is correctly installed. Note that mixing device types and speeds is strongly discouraged.

5. Show the device information for the pool:
   
   : poc:configuration storage (pool_2) verify> show
   : ID STATUS ALLOCATION DATA LOG CACHE RPM
   : 0 ok custom 0 0 0/4 1.86T
Understanding the Appliance Status

The Status section provides a summary of appliance status and configuration options. Use the following sections for conceptual and procedural information about appliance status views and related service configuration:

- **About the Oracle ZFS Storage Appliance**
“Understanding the Appliance Status” on page 86
“Dashboard Status” on page 87
“Summary of Memory Use” on page 88
“Dashboard Services” on page 89
“Dashboard Hardware” on page 90
“Disk Activity Dashboard” on page 91
“Recent Alerts” on page 93
“Dashboard CLI” on page 93
“Running the Dashboard Continuously” on page 94
“Status Dashboard Settings” on page 95
“Changing the Displayed Activity Statistics” on page 97
“Changing the Activity Thresholds” on page 97
“NDMP Status” on page 97
“NDMP States” on page 99

Dashboard Status
The Dashboard summarizes appliance status

The Status Dashboard provides links to all the main screens of the browser user interface (BUI). Over 100 visible items on the Dashboard link to associated BUI screens indicated by a border or highlighted text that appears on mouse-over. The sections that follow describe the areas of the Dashboard in detail.

The Usage area of the Dashboard provides a summary of your storage pool and main memory usage. The name of the pool appears at the top right of the Usage area. If multiple pools are configured, use the pull-down list to select the desired pool to display.

**FIGURE 5** Status Dashboard Usage

The total pool capacity is displayed at the top of this area. The Storage pie-chart details the used, available, and free space. To go to the Shares screen for the pool, click the Storage pie-chart.

**Summary of Memory Use**

The total system physical memory is displayed at the top of this area. To the left is a pie-chart showing memory usage by component. To go to the Analytics worksheet for dynamic memory usage broken down by application name, click the Memory pie-chart.
TABLE 24  Summary of Pool Usage

<table>
<thead>
<tr>
<th>Summary Pool Usage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used</td>
<td>Space used by this pool including data and snapshots.</td>
</tr>
<tr>
<td>Avail</td>
<td>Amount of physical disk space available. Space available for file data (as reported in the Shares screen) will be less than this, due to the consumption of filesystem metadata.</td>
</tr>
<tr>
<td>Free</td>
<td>Amount of space available, within the LUN capacity, less unused space that is reserved by projects and shares within a pool. Provides the free disk space available when disk space is allocated by reservation in advance and/or when LUNs are created.</td>
</tr>
<tr>
<td>Compression</td>
<td>Current compression ratio achieved by this pool. Ratio will display 1x if compression is disabled.</td>
</tr>
<tr>
<td>Dedup</td>
<td>Current data deduplication ratio achieved by this pool. Ratio will display 1x if data deduplication is disabled.</td>
</tr>
</tbody>
</table>

TABLE 25  Summary of Main Memory Usage

<table>
<thead>
<tr>
<th>Summary of main memory (RAM) usage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache</td>
<td>Bytes in use by the filesystem cache to improve performance.</td>
</tr>
<tr>
<td>Unused</td>
<td>Bytes not currently in use. After booting, this value will decrease as space is used by the filesystem cache.</td>
</tr>
<tr>
<td>Mgmt</td>
<td>Bytes in use by the appliance management software.</td>
</tr>
<tr>
<td>Other</td>
<td>Bytes in use by miscellaneous operating system software.</td>
</tr>
<tr>
<td>Kernel</td>
<td>Bytes in use by the operating system kernel.</td>
</tr>
</tbody>
</table>

Note that users need the analytics/component create+read authorization to view the memory usage. Without this authorization, the memory details do not appear on the Dashboard.

Dashboard Services

This area of the Dashboard shows the status of services on the appliance, with a light icon to show the state of each service.
Most services are green to indicate that the service is online, or grey to indicate that the service is disabled. For a reference of all possible states and icon colors, see “Browser User Interface (BUI)” on page 20 for a reference of all possible states and icon colors.

To go to the associated configuration screen, click on a service name. The Properties screen appears with configurable fields, restart, enable, and disable icons, and a link to the associated Logs screen for the service.

**Dashboard Hardware**

This area of the Dashboard shows an overview of hardware on the appliance.
If there is a known fault, the amber fault 🚫 icon appears.

To go to the Hardware screen for a detailed look at hardware state, click the name of a hardware component.

**Disk Activity Dashboard**

The activity area of the Dashboard shows graphs of eight performance statistics by default. The example in this section shows Disk operations/sec. The statistical average is plotted in blue and the maximum appears in light grey.
To go to the Analytics worksheet for an activity, click one of the four graphs (day, hour, minute, second) for the statistic you want to evaluate.

To view the average for each graph, mouse-over a graph and the average appears in the tooltip. The weather icon in the upper-left provides a report of activity according to thresholds you can customize for each statistic on the Status Settings screen.

**TABLE 26**  
**Summary of Statistic Graphs**

<table>
<thead>
<tr>
<th>Summary of Statistic Graphs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7-day graph (7d)</td>
<td>A bar chart, with each bar representing one day.</td>
</tr>
<tr>
<td>24-hour graph (24h)</td>
<td>A bar chart, with each bar representing one hour.</td>
</tr>
<tr>
<td>60-minute graph (60m)</td>
<td>A line plot, representing activity over one hour (also visible as the first one-hour bar in the 24-hour graph).</td>
</tr>
<tr>
<td>1-second graph</td>
<td>A line plot, representing instantaneous activity reporting.</td>
</tr>
</tbody>
</table>

The average for the selected plot is shown numerically above the graph. To change the average that appears, select the average you want, either 7d, 24h, or 60m.

The vertical scale of all graphs is printed on the top right, and all graphs are scaled to this same height. The height is calculated from the selected graph (plus a margin). The height will rescale based on activity in the selected graph, with the exception of utilization graphs which have a fixed height of 100 percent.

Since the height can rescale, 60 minutes of idle activity may look similar to 60 minutes of busy activity. Always check the height of the graphs before trying to interpret what they mean.

Understanding some statistics may not be obvious - you might wonder, for a particular appliance in your environment, whether 1000 NFSv3 ops/sec is considered busy or idle. This is where the 24-hour and 7-day plots can help, to provide historic data next to the current activity for comparison.

The plot height is calculated from the selected plot. By default, the 60-minute plot is selected. So, the height is the maximum activity during that 60-minute interval (plus a margin). To rescale all plots to span the highest activity during the previous 7 days, select 7d. This makes it easy to see how current activity compares to the last day or week.

The weather icon is intended to grab your attention when something is unusually busy or idle. To go to the weather threshold configuration page, click the weather icon. There is no good or bad threshold, rather the BUI provides a gradient of levels for each activity statistic. The statistics on which weather icons are based provide an approximate understanding for appliance performance that you should customize to your workload, as follows:

- Different environments have different acceptable levels for performance (latency), and so there is no one-size-fits-all threshold.
- The statistics on the Dashboard are based on operations/sec and bytes/sec, so you should use Analytics worksheets for an accurate understanding of system performance.
Recent Alerts

This section shows the last four appliance alerts. Click the box to go to the Logs screen to examine all recent alerts in detail.

Dashboard CLI

A text version of the Status > Dashboard screen is available from the CLI by typing `status dashboard`:

```plaintext
cuttlefish:> status dashboard
Storage:
  pool 0:
    Used  497G bytes
    Avail 8.58T bytes
    Free  8.43T bytes
    State online
    Compression 1x

Memory:
  Cache  30.1G bytes
  Unused 2.18G bytes
  Mgmt  343M bytes
  Other  474M bytes
  Kernel 38.9G bytes

Services:
  |   |      |   |      |
  | ad| disabled| smb| disabled|
  | dns| online   | ftp| disabled |
  | http| online   | identity| online |
  | idmap| online   | ipmp| online |
  | iscsi| online   | ldap| disabled |
  | ndmp| online   | nfs| online |
  | nis| online   | ntp| online |
  | routing| online   | scrk| maintenance |
  | smtp| online   | ssh| online |
  | tags| online   | vscan| online |
```
Running the Dashboard Continuously

You might experience browser memory issues if you leave the Dashboard screen open in a browser continuously (24x7). The browser will increase in size (memory leaks), and need to be closed and reopened. Browsers are fairly good at managing memory when browsing through different websites (and opening and closing tabs). The issue is that the Dashboard screen is left running and not closed, which opens and reopens images for the activity plots, thus degrading image rendering performance.

If you experience this problem while using Firefox, disable the memory cache as follows:

1. **Open about:config**
2. Filter on "memory"

3. Set browser.cache.memory.enable = false

Status Dashboard Settings

The Status > Settings screen enables you to customize the Status Dashboard, including the statistics that appear and thresholds that indicate activity through the weather icons.

FIGURE 10  Dashboard Settings

Use the layout tab to select the graphs that appear in the dashboard activity area, as defined in the following table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;empty&gt;</td>
<td>-</td>
<td>No graph will be displayed in this location.</td>
</tr>
<tr>
<td>SMB</td>
<td>operations/sec</td>
<td>Average number of SMB operations.</td>
</tr>
<tr>
<td>CPU</td>
<td>utilization</td>
<td>Average cycles the appliance CPUs are busy. CPU cycles includes memory wait cycles.</td>
</tr>
<tr>
<td>Disk</td>
<td>operations/sec</td>
<td>Average number of operations to the physical storage devices.</td>
</tr>
<tr>
<td>HTTP</td>
<td>operations/sec</td>
<td>Average number of HTTP operations.</td>
</tr>
<tr>
<td>iSCSI</td>
<td>operations/sec</td>
<td>Average number of iSCSI operations.</td>
</tr>
<tr>
<td>FC</td>
<td>operations/sec</td>
<td>Average number of Fibre Channel operations.</td>
</tr>
<tr>
<td>Network</td>
<td>bytes/sec</td>
<td>Average bytes/sec across all physical network interfaces.</td>
</tr>
</tbody>
</table>
Running the Dashboard Continuously

<table>
<thead>
<tr>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDMP</td>
<td>bytes/sec</td>
<td>Average NDMP network bytes.</td>
</tr>
<tr>
<td>NFSv2</td>
<td>operations/sec</td>
<td>Average number of NFSv2 operations.</td>
</tr>
<tr>
<td>NFSv3</td>
<td>operations/sec</td>
<td>Average number of NFSv3 operations.</td>
</tr>
<tr>
<td>NFSv4</td>
<td>operations/sec</td>
<td>Average number of NFSv4 operations.</td>
</tr>
<tr>
<td>FTP</td>
<td>bytes/sec</td>
<td>Average number of FTP bytes.</td>
</tr>
<tr>
<td>SFTP</td>
<td>bytes/sec</td>
<td>Average number of SFTP bytes.</td>
</tr>
</tbody>
</table>

Note that to reduce the network traffic required to refresh the Dashboard, configure some of the activity graphs as "<empty>".

Use the Thresholds screen to configure the dashboard activity weather icons. The defaults provided are based on heavy workloads and may not be suitable for your environment.

FIGURE 11  Dashboard Activity Settings

The weather icon that appears on the Dashboard is closest to the threshold value setting for the current activity - measured as a 60 second average. For example, if CPU utilization was at 41%, by default, the Cloudy weather icon would appear because its threshold is 40% (closest to the actual activity). Select the Custom radio button to configure thresholds and be sure to configure them in the order they appear on the screen.

The dashboard currently cannot be configured from the CLI. Settings saved in the BUI will apply to the dashboard that is visible from the CLI.
Changing the Displayed Activity Statistics

1. Go to the Status > Settings > Layout screen.
2. Choose the statistics you want to display on the Dashboard from the drop-down menus.
3. To save your choices, click the Apply button.

Changing the Activity Thresholds

1. Go to the Status > Settings > Thresholds screen.
2. Choose the statistic to configure from the drop-down menu.
3. Click the Custom radio button.
4. Customize the values in the list, in the order they appear. Some statistics will provide a Units drop-down, so that Kilo/Mega/Giga can be selected.
5. To save your configuration, click the Apply button.

NDMP Status

When the NDMP service has been configured and is active, the Status=>NDMP page shows the NDMP devices and recent client activity. A green indicator shows that the device is online and a gray indicator shows that the device is offline.

To resort the NDMP Device list, click on the Devices column headings. To display details about a device, double click on the device.

NDMP status is not available from the CLI.
Changing the Activity Thresholds

**TABLE 28** NDMP Status - Devices

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Type of NDMP device</td>
<td>Robot, Tape drive</td>
</tr>
<tr>
<td>Path</td>
<td>Path of the NDMP device</td>
<td>/dev/rmt/14bn</td>
</tr>
<tr>
<td>Vendor</td>
<td>Device vendor name</td>
<td>STK</td>
</tr>
<tr>
<td>Model</td>
<td>Device model name</td>
<td>T1000C</td>
</tr>
<tr>
<td>WWN</td>
<td>World Wide Name</td>
<td>50:01:04:F0:00:AC:BB:27</td>
</tr>
<tr>
<td>Serial</td>
<td>Device serial number</td>
<td>576001000203</td>
</tr>
</tbody>
</table>

**TABLE 29** NDMP Status - Recent Activity

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>NDMP backup ID</td>
<td>49</td>
</tr>
<tr>
<td>Active</td>
<td>Backup currently active</td>
<td>No</td>
</tr>
<tr>
<td>Remote Client</td>
<td>NDMP client address and port</td>
<td>192.168.1.219:4760</td>
</tr>
<tr>
<td>Authenticated</td>
<td>Shows if the client has completed authentication yet</td>
<td>Yes, No</td>
</tr>
<tr>
<td>Data State</td>
<td>See Data State</td>
<td>Active, Idle, ...</td>
</tr>
<tr>
<td>Mover State</td>
<td>See Mover State</td>
<td>Active, Idle, ...</td>
</tr>
<tr>
<td>Current Operation</td>
<td>Current NDMP operation</td>
<td>Backup, Restore, None</td>
</tr>
<tr>
<td>Progress</td>
<td>A progress bar for this backup</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Storage Area Network

NDMP States

The NDMP Data State shows the state of the backup or restore operation. Possible values are:

- **Active** - The data is being backed up or restored.
- **Idle** - Backup or restore has not yet started or has already finished.
- **Connected** - Connection is established, but backup or restore has not yet begun.
- **Halted** - Backup or restore has finished successfully or has failed or aborted.
- **Listen** - Operation is waiting to receive a remote connection.

The NDMP Mover State shows the state of the NDMP device subsystem. Examples for tape devices are:

- **Active** - Data is being read from or written to the tape.
- **Idle** - Tape operation has not yet started or has already finished.
- **Paused** - Tape has reached the end or is waiting to be changed.
- **Halted** - Read/write operation has finished successfully or has failed or aborted.
- **Listen** - Operation is waiting to receive a remote connection.

Configuring Storage Area Network

The SAN configuration page lets you connect your appliance to your SAN (Storage Area Network). A SAN is made up of three basic components:

- A client that will access the storage on the network
- A storage appliance that will provide the storage on the network
- A network that will link the client to the storage

These three components remain the same regardless of which protocol is used on the network. In some cases, the network may even be a cable between the initiator and the target, but in most cases, there is some type of switching involved.

Targets and initiators are configured by protocol. Refer to the documentation on a particular protocol ("Configuring SAN Fibre Channel" on page 104, "iSCSI" on page 187 or "Working with SRP" on page 219) for details.

Target and initiator groups define sets of targets and initiators that can be associated with LUNs. A LUN that is associated with a target group can only be seen via the targets in the group. If a LUN is not explicitly associated with a target group, it is in the *default target group* and will be accessible via all targets, regardless of protocol. Similarly, a LUN can only be seen by the initiators in the group or groups to which it belongs. If a LUN is not explicitly associated with an initiator group, it is in the *default initiator group* and can be accessed by all initiators. While using the default initiator group can be useful for evaluation purposes, its use is discouraged since it may result in exposure of the LUN to unwanted or conflicting initiators.
To avoid possible LUN conflicts when an initiator belongs to multiple groups, configure initiators within all groups before associating groups with LUNs.

To configure targets, go to the Configuration > SAN BUI page, use Fibre Channel, iSCSI, and SRP to navigate, and then configure the Ports, Initiator, and Target Groups controls.

**FIGURE 13  SAN BUI Page**

To associate a LUN, go to the Shares > Shares > Protocols page and then configure the Target Group and Initiator Group controls.
Use the configuration `san` context of the CLI to operate on targets and initiators by protocol type. Then, use the `shares` CLI context to create LUNs and associate them with target and initiator groups.

To configure SAN, use the following sections:

- “SAN Terminology” on page 102
- “Configuring SAN Fibre Channel” on page 104
- “Configuring FC Port Modes using the BUI” on page 106
- “Discovering FC Ports using the BUI” on page 107
- “Creating FC Initiator Groups using the BUI” on page 108
- “Associating a LUN with an FC Initiator Group using the BUI” on page 109
- “Changing FC Port Modes using the CLI” on page 110
- “Discovering FC Ports using the CLI” on page 110
- “Creating FC Initiator Groups using the CLI” on page 111
SAN Terminology

To configure the appliance to operate on a SAN, you should understand some basic SAN terms:

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCSI Target</td>
<td>A SCSI Target is a storage system end-point that provides a service of processing SCSI commands and I/O requests from an initiator. A SCSI Target is created by the storage system's administrator, and is identified by unique addressing methods. A SCSI Target, once configured, consists of zero or more logical units.</td>
</tr>
<tr>
<td>SCSI Initiator</td>
<td>A SCSI Initiator is an application or production system end-point that is capable of initiating a SCSI session, sending SCSI commands and I/O requests. SCSI Initiators are also identified by unique addressing methods (See SCSI Targets).</td>
</tr>
<tr>
<td>Logical Unit</td>
<td>A Logical Unit is a term used to describe a component in a storage system. Uniquely numbered, this creates what is referred to as a Logical Unit Number, or LUN. A storage system, being highly configurable, may contain many LUNS. These LUNS, when associated with one or more SCSI Targets, forms a unique SCSI device, a device that can be accessed by one or more SCSI Initiators.</td>
</tr>
<tr>
<td>iSCSI</td>
<td>Internet SCSI, a protocol for sharing SCSI based storage over IP networks.</td>
</tr>
<tr>
<td>iSER</td>
<td>iSCSI Extension for RDMA, a protocol that maps the iSCSI protocol over a network that provides</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FC</td>
<td>Fibre Channel, a protocol for sharing SCSI based storage over a storage area network (SAN), consisting of fiber-optic cables, FC switches and HBAs.</td>
</tr>
<tr>
<td>SRP</td>
<td>SCSI RDMA Protocol, a protocol for sharing SCSI based storage over a network that provides RDMA services (i.e. InfiniBand).</td>
</tr>
<tr>
<td>IQN</td>
<td>An iSCSI qualified name, the unique identifier of a device in an iSCSI network. iSCSI uses the form iqn. date.authority:uniqueid for IQNs. For example, the appliance may use the IQN: iqn.1986-03.com.sun:02:c7824a5b-f3ea-6038-c79d-ca443337d92c to identify one of its iSCSI targets. This name shows that this is an iSCSI device built by a company registered in March of 1986. The naming authority is just the DNS name of the company reversed, in this case, &quot;com.sun&quot;. Everything following is a unique ID that Sun uses to identify the target.</td>
</tr>
<tr>
<td>Target portal</td>
<td>When using the iSCSI protocol, the target portal refers to the unique combination of an IP address and TCP port number by which an initiator can contact a target.</td>
</tr>
<tr>
<td>Target portal group</td>
<td>When using the iSCSI protocol, a target portal group is a collection of target portals. Target portal groups are managed transparently; each network interface has a corresponding target portal group with that interface's active addresses. Binding a target to an interface advertises that iSCSI target using the portal group associated with that interface.</td>
</tr>
<tr>
<td>CHAP</td>
<td>Challenge-handshake authentication protocol, a security protocol which can authenticate a target to an initiator, an initiator to a target, or both.</td>
</tr>
<tr>
<td>RADIUS</td>
<td>A system for using a centralized server to perform CHAP authentication on behalf of storage nodes.</td>
</tr>
<tr>
<td>Target group</td>
<td>A set of targets. LUNs are exported over all the targets in one specific target group.</td>
</tr>
<tr>
<td>Initiator group</td>
<td>A set of initiators. When an initiator group is associated with a LUN, only initiators from that group may access the LUN.</td>
</tr>
<tr>
<td>Target</td>
<td>A storage system end-point that provides a service of processing SCSI commands and I/O requests from an initiator. A target is created by the storage system administrator, and is identified by unique addressing methods. A target, once configured, consists of zero or more logical units.</td>
</tr>
<tr>
<td>Initiator</td>
<td>An application or production system end-point that is capable of initiating a SCSI session, sending SCSI</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>commands and I/O requests. Initiators are also identified by unique addressing methods.</td>
<td></td>
</tr>
</tbody>
</table>

Each LUN has several properties which control how the volume is exported. See “Protocols” on page 314 for more information.

## Configuring SAN Fibre Channel

Fibre Channel (FC) is a gigabit-speed networking technology used nearly exclusively as a transport for SCSI. FC is one of several block protocols supported by the appliance; to share LUNs via FC, the appliance must be equipped with one or more optional FC cards.

By default, all FC ports are configured to be in target mode. If the appliance is used to connect to a tape SAN for backup, one or more ports must be configured in initiator mode. To configure a port for initiator mode, the appliance must be reset. Multiple ports can be configured for initiator mode simultaneously.

Each FC port is assigned a World Wide Name (WWN), and, as with other block protocols, FC targets may be grouped into SAN target and initiator groups, allowing port bandwidth to be dedicated to specific LUNs or groups of LUNs. Once an FC port is configured as a target, the remotely discovered ports can be examined and verified.

Refer to the Implementing Fibre Channel SAN Boot with Oracle’s Sun ZFS Storage Appliance white paper at [http://www.oracle.com/technetwork/articles/servers-storage-admin/fbsanboot-365291.html](http://www.oracle.com/technetwork/articles/servers-storage-admin/fbsanboot-365291.html) for details on FC SAN boot solutions using the Oracle ZFS Storage Appliance.

In a cluster, initiators will have two paths (or sets of paths) to each LUN: one path (or set of paths) will be to the head that has imported the storage associated with the LUN; the other path (or set of paths) will be to that head’s clustered peer. The first path (or set of paths) is active; the second path (or set of paths) is standby. In the event of a takeover, the active paths will become unavailable, and the standby paths will (after a short time) be transitioned to be active, after which I/O will continue. This approach to multipathing is known as asymmetric logical unit access (ALUA) and, when coupled with an ALUA-aware initiator, allows cluster takeover to be transparent to higher-level applications.

Initiators are identified by their WWN. As with other block protocols, aliases can be created for initiators. To aid in creating aliases for FC initiators, a WWN can be selected from the WWNs of discovered ports. Also as with other block protocols, initiators can be collected into groups. When a LUN is associated with a specific initiator group, the LUN will only be visible to initiators in the group. In most FC SANs, LUNs will always be associated with the initiator group that corresponds to the system(s) for which the LUN has been created.

The appliance is an ALUA-compliant array. Properly configuring an FC initiator in an ALUA environment requires an ALUA-aware driver and may require initiator-specific tuning. See
"Oracle ZFS Storage Appliance: How to set up Client Multipathing" (Doc ID 1628999.1) for more information.

FC performance can be observed via Analytics, whereby one can breakdown operations or throughput by initiator, target, or LUN:

**FIGURE 15** FC Performance

For operations, one can also breakdown by offset, latency, size and SCSI command, allowing one to understand not just the *what* but the *how* and *why* of FC operations.

The appliance has been designed to utilize a global set of resources to service LUNs on each head. It is therefore not generally necessary to restrict queue depths on clients as the FC ports in the appliance can handle a large number of concurrent requests. Even so, there exists the remote possibility that these queues can be overrun, resulting in SCSI transport errors. Such queue overruns are often associated with one or more of the following:
- Overloaded ports on the front end - too many hosts associated with one FC port and/or too many LUNs accessed through one FC port
- Degraded appliance operating modes, such as a cluster takeover in what is designed to be an active-active cluster configuration

While the possibility of queue overruns is remote, it can be eliminated entirely if one is willing to limit queue depth on a per-client basis. To determine a suitable queue depth limit, one should take the number of target ports multiplied by the maximum concurrent commands per port (2048) and divide the product by the number of LUNs provisioned. To accommodate degraded operating modes, one should sum the number of LUNs across cluster peers to determine the number of LUNs, but take as the number of target ports the minimum of the two cluster peers. For example, in an active-active 7420 dual-headed cluster with one head having 2 FC ports and 100 LUNs and the other head having 4 FC ports and 28 LUNs, one should take the pessimal maximum queue depth to be two ports times 2048 commands divided by 100 LUNs plus 28 LUNs -- or 32 commands per LUN.

Tuning the maximum queue depth is initiator specific, but on Solaris, this is achieved by adjusting the global variable `ssd_max_throttle`.

To troubleshoot link-level issues such as broken optics or a poorly seated cable, look at the error statistics for each FC port. If any number is either significantly non-zero or increasing, that may be an indicator that link-level issues have been encountered, and that link-level diagnostics should be performed.

**Configuring FC Port Modes using the BUI**

To make use of FC ports, set them to Target mode on the Configuration > SAN screen of the BUI, using the drop-down menu shown in the screenshot below. You must have root permissions to perform this action. Note that in a cluster configuration, you will set ports to Target mode on each head node separately.
After setting desired ports to Target, click the Apply button. A confirmation message will appear notifying you that the appliance will reboot immediately. Confirm that you want to reboot.

When the appliance boots, the active FC targets appear with the icon and, on mouse-over, the move icon appears.

**Discovering FC Ports using the BUI**

Click the info icon to view the Discovered Ports dialog where you can troubleshoot link errors. In the Discovered Ports dialog, click a WWN in the list to view associated link errors.
Creating FC Initiator Groups using the BUI

Create and manage initiator groups on the Initiators screen. Click the add button to view unaliased ports. Click a WWN in the list to add a meaningful alias in the Alias field.
On the Initiators page, drag initiators to the FC Initiator Groups list to create new groups or add to existing groups.

**FIGURE 17** FC Initiator Groups List

Click the Apply button to commit the new Initiator Group. Now you can create a LUN that has exclusive access to the client initiator group.

**Associating a LUN with an FC Initiator Group using the BUI**

To create the LUN, roll-over the initiator group and click the add LUN icon. The Create LUN dialog appears with the associated initiator group selected. Set the name and size and click Apply to add the LUN to the storage pool.
Changing FC Port Modes using the CLI

To change FC port modes, use the following CLI commands:

```
dory:configuration san fc targets> set targets="wwn.2101001B32A11639"
  targets = wwn.2101001B32A11639 (uncommitted)
dory:configuration san fc targets> commit
```

Discovering FC Ports using the CLI

To discover FC Ports, use the following CLI commands:

```
dory:configuration san fc targets> show
Properties:
```
Creating FC Initiator Groups using the CLI

Targets:

<table>
<thead>
<tr>
<th>NAME</th>
<th>MODE</th>
<th>WWN</th>
<th>PORT</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>target-000</td>
<td>target</td>
<td>wwn.2100001B32811639</td>
<td>PCIe 5: Port 1</td>
<td>4 Gbit/s</td>
</tr>
<tr>
<td>target-001</td>
<td>initiator</td>
<td>wwn.2101001B32A11639</td>
<td>PCIe 5: Port 2</td>
<td>0 Gbit/s</td>
</tr>
<tr>
<td>target-002</td>
<td>initiator</td>
<td>wwn.2100001B32812239</td>
<td>PCIe 2: Port 1</td>
<td>0 Gbit/s</td>
</tr>
<tr>
<td>target-003</td>
<td>target</td>
<td>wwn.2101001B32A12239</td>
<td>PCIe 2: Port 2</td>
<td>0 Gbit/s</td>
</tr>
</tbody>
</table>

dory:configuration san fc targets> select target-000
dory:configuration san fc targets target-000> show

Properties:

- wwn = wwn.2100001B32811639
- port = PCIe 5: Port 1
- mode = target
- speed = 4 Gbit/s

Ports:

<table>
<thead>
<tr>
<th>PORT</th>
<th>WWN</th>
<th>ALIAS</th>
<th>MANUFACTURER</th>
</tr>
</thead>
<tbody>
<tr>
<td>port-000</td>
<td>wwn.2100001B3281A339</td>
<td>longjaw-1</td>
<td>QLogic Corporation</td>
</tr>
<tr>
<td>port-001</td>
<td>wwn.2101001B32A1A339</td>
<td>longjaw-2</td>
<td>QLogic Corporation</td>
</tr>
<tr>
<td>port-002</td>
<td>wwn.2100001B32A1AC39</td>
<td>thicktail-1</td>
<td>QLogic Corporation</td>
</tr>
<tr>
<td>port-003</td>
<td>wwn.2101001B32A1AC39</td>
<td>thicktail-2</td>
<td>QLogic Corporation</td>
</tr>
<tr>
<td>port-004</td>
<td>wwn.2100001B3281E339</td>
<td>&lt;none&gt;</td>
<td>QLogic Corporation</td>
</tr>
<tr>
<td>port-005</td>
<td>wwn.2101001B32A1E339</td>
<td>&lt;none&gt;</td>
<td>QLogic Corporation</td>
</tr>
</tbody>
</table>

▼ Creating FC Initiator Groups using the CLI

- To create FC initiator groups, use the following CLI commands:


dory:configuration san fc initiators> create
dory:configuration san fc initiators (uncommitted)> set name=lefteye
dory:configuration san fc initiators (uncommitted)> set initiators=wwn.2101001B32A1AC39, wwn.2100001B3281AC39
dory:configuration san fc initiators (uncommitted)> commit
dory:configuration san fc initiators> list

GROUP     NAME
---       ----
group-001 lefteye
       +--> INITIATORS
               wwn.2101001B32A1AC39
               wwn.2100001B3281AC39
Associating a LUN with an FC Initiator Group using the CLI

The following example demonstrates creating a LUN called lefty and associating it with the fera initiator group.

- To associate a LUN with an FC initiator group, use the following CLI commands:

```
dory:shares default> lun lefty
dory:shares default/lefty (uncommitted)> set volsize=10
    volsize = 10 (uncommitted)
dory:shares default/lefty (uncommitted)> set initiatorgroup=fera
    initiatorgroup = default (uncommitted)
dory:shares default/lefty (uncommitted)> commit
```

Scripting Aliases for Initiators and Initiator Groups using the CLI

Refer to “CLI Usage” on page 32 and “Simple CLI Scripting and Batching Commands” on page 40 for information about how to modify and use the following example script.

- To script aliases for initiators and initiator groups, use the following CLI commands:

```bash
script
/*
 * This script creates both aliases for initiators and initiator
groups, as specified by the below data structure. In this
 * particular example, there are five initiator groups, each of
 * which is associated with a single host (thicktail, longjaw, etc.),
 * and each initiator group consists of two initiators, each of which
 * is associated with one of the two ports on the FC HBA. (Note that
 * there is nothing in the code that uses this data structure that
 * assumes the number of initiators per group.)
 */
groups = {
    thicktail: {
        'thicktail-1': 'wwn.2100001b3281ac39',
        'thicktail-2': 'wwn.2101001b32a1ac39'
    },
    longjaw: {
        'longjaw-1': 'wwn.2100001b3281a339',
        'longjaw-2': 'wwn.2101001b32a1a339'
    },
```
Scripting Aliases for Initiators and Initiator Groups using the CLI

```javascript
for (group in groups) {
    initiators = [];
    for (initiator in groups[group]) {
        printf('Adding %s for %s ... ',
            groups[group][initiator], initiator);
        try {
            run('select alias=' + initiator);
            printf('(already exists)
');
            run('cd ..');
        } catch (err) {
            if (err.code != EAKSH_ENTITY_BADSELECT)
                throw err;
            run('create');
            set('alias', initiator);
            set('initiator', groups[group][initiator]);
            run('commit');
            printf('done
');
        }
        run('select alias=' + initiator);
        initiators.push(get('initiator'));
        run('cd ..');
    }
    printf('Creating group for %s ... ', group);
    run('groups');
    try {
        run('select name=' + group);
        printf('(already exists)
');
        run('cd ..');
    } catch (err) {
        if (err.code != EAKSH_ENTITY_BADSELECT)
            throw err;
        run('create');
        set('name', group);
        run('set initiators=' + initiators);
        run('commit');
        printf('done
');
    }
    run('cd ..');
}
```
Configuring SAN iSCSI

Internet SCSI is one of several block protocols supported by the appliance for sharing SCSI based storage.

When using the iSCSI protocol, the target portal refers to the unique combination of an IP address and TCP port number by which an initiator can contact a target.

When using the iSCSI protocol, a target portal group is a collection of target portals. Target portal groups are managed transparently; each network interface has a corresponding target portal group with that interface's active addresses. Binding a target to an interface advertises that iSCSI target using the portal group associated with that interface.

**Note** - Multiple connections per session are not supported.

An IQN (iSCSI qualified name) is the unique identifier of a device in an iSCSI network. iSCSI uses the form iqn.date.authority:uniqueid for IQNs. For example, the appliance may use the IQN: iqn.1986-03.com.sun:02:c7824a5b-f3ea-6038-c79d-ca443337d92c to identify one of its iSCSI targets. This name shows that this is an iSCSI device built by a company registered in March of 1986. The naming authority is just the DNS name of the company reversed, in this case, "com.sun". Everything following is a unique ID that Oracle uses to identify the target.

<table>
<thead>
<tr>
<th><strong>TABLE 31</strong></th>
<th>iSCSI Target Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target Property</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Target IQN</td>
<td>The IQN for this target. The IQN can be manually specified or auto-generated.</td>
</tr>
<tr>
<td>Alias</td>
<td>A human-readable nickname for this target.</td>
</tr>
<tr>
<td>Authentication mode</td>
<td>One of None, CHAP, or RADIUS.</td>
</tr>
<tr>
<td>CHAP name</td>
<td>If CHAP authentication is used, the CHAP username.</td>
</tr>
<tr>
<td>CHAP secret</td>
<td>If CHAP authentication is used, the CHAP secret.</td>
</tr>
<tr>
<td>Network interfaces</td>
<td>The interfaces whose target portals are used to export this target.</td>
</tr>
</tbody>
</table>

In addition to those properties, the BUI indicates whether a target is online or offline:

<table>
<thead>
<tr>
<th><strong>TABLE 32</strong></th>
<th>Target Status Icons</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Icon</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>![Online Icon]</td>
<td>Target is online</td>
</tr>
<tr>
<td>![Offline Icon]</td>
<td>Target is offline</td>
</tr>
</tbody>
</table>
On clustered platforms, targets which have at least one active interface on that cluster node will be online. Take care when assigning interfaces to targets; a target may be configured to use portal groups on disjoint head nodes. In that situation, the target will be online on both heads yet will export different LUNs depending on the storage owned by each head node. As network interfaces migrate between cluster heads as part of takeover/failback or ownership changes, iSCSI targets will move online and offline as their respective network interfaces are imported and exported.

Targets which are bound to an IPMP interface will be advertised only via the addresses of that IPMP group. That target will not be reachable via that group's test addresses. Targets bound to interfaces built on top of a LACP aggregation will use the address of that aggregation. If a LACP aggregation is added to an IPMP group, a target can no longer use that aggregation's interface, as that address will become an IPMP test address.

## Configuring SAN iSCSI Initiators

iSCSI initiators have the following configurable properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiator IQN</td>
<td>The IQN for this initiator.</td>
</tr>
<tr>
<td>Alias</td>
<td>A human-readable nickname for this initiator.</td>
</tr>
<tr>
<td>Use CHAP</td>
<td>Enables or disables CHAP authentication</td>
</tr>
<tr>
<td>CHAP name</td>
<td>If CHAP authentication is used, the CHAP username.</td>
</tr>
<tr>
<td>CHAP secret</td>
<td>If CHAP authentication is used, the CHAP secret.</td>
</tr>
</tbody>
</table>

When planning your iSCSI client configuration, you'll need the following information:

- What initiators (and their IQNs) will be accessing the SAN?
- If you plan on using CHAP authentication, what CHAP credentials does each initiator use?
- How many iSCSI disks (LUNs) are required, and how big should they be?
- Do the LUNs need to be shared between multiple initiators?

To allow the Appliance to perform CHAP authentication using RADIUS, the following pieces of information must match:

- The Appliance must specify the address of the RADIUS server and a secret to use when communicating with this RADIUS server
- The RADIUS server (e.g. in its clients file) must have an entry giving the address of this Appliance and specifying the same secret as above
- The RADIUS server (e.g. in its users file) must have an entry giving the CHAP name and matching CHAP secret of each initiator
Creating an Analytics Worksheet using the BUI

To create an analytics worksheet for observing operations by initiator, complete the following:

1. **Go to the Analytics screen.**
2. **Click the add icon for Add Statistic.** A menu of all statistics appears.
3. **Select iSCSI operations > Broken down by initiator under the Protocols section of the menu.** A graph of the current operations by initiator appears.
4. **To observe more detailed analytics, select the initiator from the field to the left of the graph and click the icon.** A menu of detailed analytics appears.

### Configuring SAN iSER Targets

In the BUI, iSER targets are managed as iSCSI targets on the Configuration > SAN screen.

1. **To configure ibp(x) interfaces,** select the ibp(x) interface (or ipmp) you want, and drag it to the Datalinks list to create the datalink on the Configuration > Network screen.
2. **Drag the Datalink to the Interfaces list** to create a new interface.

---

- If the initiator uses its IQN name as its CHAP name (the recommended configuration) then the Appliance does not need a separate Initiator entry for each Initiator box -- the RADIUS server can perform all authentication steps.

- If the initiator uses a separate CHAP name, then the Appliance must have an Initiator entry for that initiator that specifies the mapping from IQN name to CHAP name. This Initiator entry does NOT need to specify the CHAP secret for the initiator.

For tips on troubleshooting common iSCSI misconfiguration, see “iSCSI” on page 187.

iSCSI performance can be observed via Analytics, whereby one can breakdown operations or throughput by initiator, target, or LUN.
3. To create an iSER target, on the Configuration > SAN page, click the iSCSI Targets link.

4. To add a new iSER target with an alias, click the add icon.

5. To create a target group, drag the target you just created to the iSCSI Target Group list.

6. To create an initiator, click the Initiator link and then click the iSCSI initiators link.

7. To add a new initiator, click the add icon.
8. Enter the Initiator IQN and an alias and click OK. Creating an initiator group is optional but if you don’t create a group, the LUN associated with the target will be available to all initiators.

9. To create a group, drag the initiator to the iSCSI Initiator Groups list.

10. To create a LUN, on the Shares page, click LUN.

11. Click the + add icon and associate the new LUN with target or initiator groups you created already using the Target Group and Initiator Groups menu.
Adding an iSCSI Target with an Auto-generated IQN using the CLI

To add an iSCSI target with an auto-generated IQN, use the following CLI commands:

ahi:configuration san iscsi targets> create
ahi:configuration san iscsi targets target (uncommitted)> set alias="Target 0"
ahi:configuration san iscsi targets target (uncommitted)> set auth=none
ahi:configuration san iscsi targets target (uncommitted)> set interfaces=igb1
ahi:configuration san iscsi targets target (uncommitted)> commit
ahi:configuration san iscsi targets> list

TARGET   ALIAS
  target-000 Target 0
    <+-- IQN
      iqn.1986-03.com.sun:02:da0161f-9f5d-e01a-b5c5-e1efa958416
Adding an iSCSI Target with a Specific IQN and RADIUS Authentication using the CLI

To add an iSCSI target with a specific IQN and RADIUS authentication, use the following CLI commands:

```
ahi:configuration san iscsi targets> create
ahi:configuration san iscsi targets target (uncommitted)> set alias="Target 1"
ahi:configuration san iscsi targets target (uncommitted)>
  set iqn=iqn.2001-02.com.acme:12345
ahi:configuration san iscsi targets target (uncommitted)> set auth=radius
ahi:configuration san iscsi targets target (uncommitted)> set interfaces=igb1
ahi:configuration san iscsi targets target (uncommitted)> commit
ahi:configuration san iscsi targets> list
```

```
TARGET     ALIAS
| target-000 Target 0
  | +--> IQN
  |  iqn.1986-03.com.sun:02:daf0161f-9f5d-e01a-b5c5-e1efa9578416
| target-001 Target 1
  | +--> IQN
  |  iqn.2001-02.com.acme:12345
```

Adding an iSCSI Initiator with CHAP Authentication using the CLI

To add an iSCSI initiator with CHAP authentication, use the following CLI commands:

```
ahi:configuration san iscsi initiators> create
ahi:configuration san iscsi initiators initiator (uncommitted)>
  set initiator=iqn.2001-02.com.acme:initiator12345
ahi:configuration san iscsi initiators initiator (uncommitted)> set alias="Init 0"
ahi:configuration san iscsi initiators initiator (uncommitted)>
  set chapuser=thisismychapuser
ahi:configuration san iscsi initiators initiator (uncommitted)>
  set chapsecret=123456789012abc
ahi:configuration san iscsi initiators initiator (uncommitted)> commit
ahi:configuration san iscsi initiators> list
```

```
NAME          ALIAS
| initiator-000 Init 0
  | +--> INITIATOR
  |  iqn.2001-02.com.acme:initiator12345
```
Adding an iSCSI Target Group using the CLI

- To add an iSCSI target group, use the following CLI commands:

```
ahi:configuration san iscsi targets groups> create
ahi:configuration san iscsi targets group (uncommitted)> set name=tg0
ahi:configuration san iscsi targets group (uncommitted)>
  set targets=iqn.2001-02.com.acme:12345,
  iqn.1986-03.com.sun:02:da0161f-9f5d-e01a-b5c5-e1efa9578416
ahi:configuration san iscsi targets group (uncommitted)> commit
ahi:configuration san iscsi targets groups> list
GROUP     NAME
group-000 tg0
  |-> TARGETS
    iqn.2001-02.com.acme:12345
    iqn.1986-03.com.sun:02:da0161f-9f5d-e01a-b5c5-e1efa9578416
```

Adding an iSCSI Initiator Group using the CLI

- To add an iSCSI initiator group, use the following CLI commands:

```
ahi:configuration san iscsi initiators groups> create
ahi:configuration san iscsi initiators group (uncommitted)> set name=ig0
ahi:configuration san iscsi initiators group (uncommitted)>
  set initiators=iqn.2001-02.com.acme:initiator12345
ahi:configuration san iscsi initiators group (uncommitted)> commit
ahi:configuration san iscsi initiators groups> list
GROUP     NAME
group-000 ig0
  |-> INITIATORS
    iqn.2001-02.com.acme:initiator12345
```

Configuring SAN SRP

SCSI RDMA Protocol, is a protocol supported by the appliance for sharing SCSI based storage over a network that provides RDMA services (i.e. InfiniBand).

SRP ports are shared with other IB port services such as IPoIB and RDMA. The SRP service may only operate in target mode. SRP targets have the following configurable properties.
### TABLE 34  
**SRP Target Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target EUI</td>
<td>The Extended Unique Identifier (EUI) for this target. The EUI is automatically assigned by the system and is equal to the HCA GUID over which the SRP port service is running.</td>
</tr>
<tr>
<td>Alias</td>
<td>A human-readable nickname for this target.</td>
</tr>
</tbody>
</table>

In addition to those properties, the BUI indicates whether a target is online or offline:

### TABLE 35  
**SRP Target Status Icons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Online Icon" /></td>
<td>Target is online</td>
</tr>
<tr>
<td><img src="image" alt="Offline Icon" /></td>
<td>Target is offline</td>
</tr>
</tbody>
</table>

On clustered platforms, peer targets should be configured into the same target group for highly available (multi-pathed) configurations. SRP multipathed I/O is an initiator-side configuration option.

SRP initiators have the following configurable properties.

### TABLE 36  
**SRP Initiator Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiator EUI</td>
<td>The EUI for this initiator.</td>
</tr>
<tr>
<td>Alias</td>
<td>A human-readable nickname for this initiator.</td>
</tr>
</tbody>
</table>

SRP performance can be observed via Analytics, whereby one can breakdown operations or throughput by initiator or target.

---

**Configuring SRP Target using the BUI**

This procedure describes the steps for configuring SRP targets.

1. **Connect HCA ports to IB interfaces.**
2. **The targets are automatically discovered by the appliance.**
3. **To create the target group, go to the Configuration > SAN screen.**
4. **Click the Target link and then click SRP targets.**
5. The SRP targets page appears.

6. To create the target group, use the move icon to drag a target to the Target Groups list.

7. Click Apply.

8. (Optional) To create an initiator and initiator group on the Initiator screen, click the icon, collect GUID from initiator, assign it a name, and drag it to initiator group.

9. To create a LUN and associate it with the SRP target and initiators you created in the previous steps, go to the Shares screen.

10. Click the LUN link and then click the LUN icon. Use the Target Group and Initiator Group menus on the Create LUN dialog to select the SRP groups to associate with the LUN.

### Configuring SRP Targets Using the CLI

The following example demonstrates how to create an SRP target group named targetSRPgroup using the CLI configuration san targets srp groups context:

- To configure SRP targets, use the following CLI commands:

```bash
swaller:configuration san targets srp groups> create
swaller:configuration san targets srp group (uncommitted)> set name=targetSRPgroup
    name = targetSRPgroup (uncommitted)
swaller:configuration san targets srp group (uncommitted)> set targets=eui.0002C903000489A4
    targets = eui.0002C903000489A4 (uncommitted)
swaller:configuration san targets srp group (uncommitted)> commit
swaller:configuration san targets srp groups> list

GROUP     NAME
group-000 targetSRPgroup
+-> TARGETS
    eui.0002C903000489A4
```

**Example 1** Creating a LUN associated with the Target SRP Group using the CLI

The following example shows how to create a LUN and associate it with the targetSRPgroup using the CLI shares CLI context:
swallower:shares default> lun mylun
swallower:shares default/mylun (uncommitted)> set targetgroup=targetSRPgroup
    targetgroup = targetSRPgroup (uncommitted)
swallower:shares default/mylun (uncommitted)> set volsize=10
    volsize = 10 (uncommitted)
swallower:shares default/mylun (uncommitted)> commit
swallower:shares default> list

Filesystems:
NAME           SIZE   MOUNTPOINT
  test          38K     /export/test

LUNs:
NAME    SIZE     GUID
  mylun   10G    600144F0E9D19FFB00004B82DF490001

Configuring Users

This section describes users who may administer the appliance, roles to manage authorizations granted to users, and how to add them to the system using the BUI or CLI.

Users can either be:

- **Local users** - Where all account information is saved on the Oracle ZFS Storage Appliance.
- **Directory users** - Where existing NIS or LDAP accounts are used and supplemental authorization settings are saved on the appliance. Access to the appliance must be explicitly granted to existing NIS/LDAP users, who can then log in to and administer the appliance. Access cannot be granted by default.

Although local users are supported for data services, there are several things to keep in mind.

- For local users, you have no control over the UIDs. This is a problem for NFSv3 using anything else and NFSv4 using AUTH_SYS.
- Local groups are not supported.
- Defining a local user for data purposes also allows the local user to log into the administrative interface.

Users are granted privileges by assigning them custom roles.

A role is a collection of privileges that can be assigned to users. It may be desirable to create administrator and operator roles, with different authorization levels. Staff members may be assigned any role that is suitable for their needs, without assigning unnecessary privileges.

The use of roles is considered to be much more secure than the use of shared administrator passwords, for example, giving everyone the root password. Roles restrict users to necessary authorizations only, and also attribute their actions to their individual username in the log.

To configure user users, use the following sections:

- “User Authorizations” on page 125
- “Managing User Properties” on page 126
User Authorizations

Authorizations allow users to perform specific tasks, such as creating shares, rebooting the appliance, and updating the system software. Authorizations are grouped into Scopes, and each scope may have a set of optional filters to narrow the scope of the authorization. For example, rather than an authorization to restart all services, a filter can be used so that this authorization can restart the HTTP service only.

The following table shows the available scopes:

<table>
<thead>
<tr>
<th>Scope BUI</th>
<th>Scope CLI</th>
<th>Example Authorization</th>
<th>Example Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Directory</td>
<td>ad</td>
<td>Join an Active Directory domain</td>
<td>Domain name</td>
</tr>
<tr>
<td>Alerts</td>
<td>alert</td>
<td>Configure alert filters and thresholds</td>
<td></td>
</tr>
<tr>
<td>Analytics</td>
<td>stat</td>
<td>Read a statistic with this drilldown present</td>
<td>Drilldowns</td>
</tr>
<tr>
<td>Clustering</td>
<td>cluster</td>
<td>Failback resources to a cluster peer</td>
<td></td>
</tr>
<tr>
<td>Datasets</td>
<td>dataset</td>
<td>Manage aspects of Analytics datasets</td>
<td>Configure</td>
</tr>
<tr>
<td>Hardware</td>
<td>hardware</td>
<td>Online and offline disks</td>
<td></td>
</tr>
<tr>
<td>Keystores</td>
<td>keystore</td>
<td>Configure keystores.</td>
<td></td>
</tr>
<tr>
<td>Networking</td>
<td>net</td>
<td>Configure networking devices, datalinks, and interfaces</td>
<td></td>
</tr>
<tr>
<td>Projects and shares</td>
<td>nas</td>
<td>Change general properties of projects and shares</td>
<td>Pool, project, share</td>
</tr>
<tr>
<td>Roles</td>
<td>role</td>
<td>Configure authorizations for a role</td>
<td>Role name</td>
</tr>
<tr>
<td>SAN</td>
<td>stmf</td>
<td>Configure authorizations for SAN</td>
<td></td>
</tr>
</tbody>
</table>
Browse the scopes in the BUI to see what other authorizations exist. There are currently over fifty different authorizations available, and additional authorizations may be added in future appliance software updates.

## Managing User Properties

The BUI Users page lists both users and groups, along with buttons for administration. Mouse-over an entry to expose its clone, edit and destroy buttons. Double-click an entry to view its edit screen. The buttons are as follows:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add new user/role. This will display a new dialog where the required properties may be entered.</td>
<td>Add new user/role. This will display a new dialog where the required properties may be entered.</td>
</tr>
<tr>
<td>Displays a search box. Enter a search string and hit enter to search the user/role lists for that text, and only display entries that match. Click this icon again or &quot;Show All&quot; to return to the full listings.</td>
<td>Displays a search box. Enter a search string and hit enter to search the user/role lists for that text, and only display entries that match. Click this icon again or &quot;Show All&quot; to return to the full listings.</td>
</tr>
<tr>
<td>Clone user/role. Add a new user/role starting with fields based on the values from this entry</td>
<td>Clone user/role. Add a new user/role starting with fields based on the values from this entry</td>
</tr>
<tr>
<td>Edit user/role</td>
<td>Edit user/role</td>
</tr>
<tr>
<td>Remove user/role/authorization</td>
<td>Remove user/role/authorization</td>
</tr>
</tbody>
</table>

All of the following properties may be set when adding a user, and a subset of these when editing a user:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Directory (access credentials from NIS or LDAP), or Local (save user on this appliance)</td>
</tr>
</tbody>
</table>
Adding an Administrator using the BUI

Adding an Administrator using the BUI

The following properties can be set when managing roles:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>Unique name for user</td>
</tr>
<tr>
<td>Full Name</td>
<td>User description</td>
</tr>
<tr>
<td>Password/Confirm</td>
<td>For Local users, type the initial password in both of these fields</td>
</tr>
<tr>
<td>Require session annotation</td>
<td>If enabled, when users login to the appliance they must provide a text description of the purpose of their login. This annotation may be used to track work performed for requests in a ticketing system, and the ticket ID can be used as the session annotation. The session annotation appears in the log.</td>
</tr>
<tr>
<td>Kiosk user</td>
<td>If enabled, the user will only be able to view the screen in the “Kiosk screen” setting. This may be used to restrict a user to only see the dashboard, for example. A kiosk user will not be able to access the appliance via the CLI.</td>
</tr>
<tr>
<td>Kiosk screen</td>
<td>Screen that this kiosk user is restricted to, if “Kiosk user” is enabled</td>
</tr>
<tr>
<td>Roles</td>
<td>The roles possessed by this user</td>
</tr>
<tr>
<td>Exceptions</td>
<td>These authorizations are excluded from those normally available due to the selected roles</td>
</tr>
</tbody>
</table>

The following properties can be set when managing roles:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the role as it will be shown in lists</td>
</tr>
<tr>
<td>Description</td>
<td>Verbose description of role if desired</td>
</tr>
<tr>
<td>Authorizations</td>
<td>Authorizations for this role</td>
</tr>
</tbody>
</table>

1. Check that an appropriate administrator role is listed in the Roles list. If not, add a role (see separate task).

2. Click the add icon next to Users.

3. Set user properties.

4. Click the checkbox for the administrator role.

5. Click the Add button at the top of the dialog. The new user appears in the Users list.
Adding a Role using the BUI

1. Click the add icon next to Roles.
2. Set the name of the role, and description.
3. Add authorizations to the role (see separate task).
4. Click the Add button at the top of the dialog. The new role appears in the Roles list.

Adding Authorizations to a Role using the BUI

1. Select "Scope". If filters are available for this scope, they will appear beneath the Scope selector.
2. Select filters if appropriate.
3. Click the checkbox for all authorizations you wish to add.
4. Click the Add button in the Authorization section. The authorizations will be added to the bottom list of the dialog box.

Deleting Authorizations from a Role using the BUI

1. Mouse-over the role in the Roles list, and click the edit icon.
2. Mouse-over the authorization in the bottom list, and click the trash icon on the right.
3. Click the Apply button at the top of the dialog.

Adding a User Who can Only View the Dashboard using the BUI

1. Add either a Directory or Local user (see separate task).
2. Set Kiosk mode to true, and check that the Kiosk screen is set to "status/dashboard".

3. The user should now be able to login, but only view the dashboard.

▼ Adding a Role using the CLI

1. Go to configuration roles.

2. Type role followed by the role name you wish to create.

3. Set the description, then type commit to commit the role.

4. Add authorizations to the role (see separate task).

**Example 2** CLI Example for Adding a Role

This CLI Example creates the role "webadmin".

```
caji:> configuration roles
caji:configuration roles> role webadmin
caji:configuration roles webadmin (uncommitted)> set
description="web server administrator"
description = web server administrator (uncommitted)
caji:configuration roles webadmin (uncommitted)> commit
caji:configuration roles> show
Roles:

NAME        DESCRIPTION
basic        Basic administration
webadmin     web server administrator
```

▼ Adding an Administrator using the CLI

1. Go to configuration roles.

2. Type show. Find a role with appropriate administration authorizations by running select on each role and then authorizations show. If an appropriate role does not exist, start by creating the role (see separate task).

3. Go to configuration users.
4. For Directory users (NIS, LDAP), type `netuser` followed by the existing username you wish to add. For Local users, type `user` followed by the username you wish to add; then type `show` to see the properties that need to be set. Type `set` then, then type `commit`.

5. At this point you have a created user, but haven't customized all their properties yet. Type `select` followed by their username.

6. Now type `show` to see the full list of preferences. You can now add roles and authorization exceptions and set appliance preferences (See “Setting Oracle ZFS Storage Appliance Preferences” on page 132).

Example 3 CLI Example for Creating a User and Assigning a Role

This CLI Example creates the user "brendan" and assigns the role "webadmin":

```bash
caji:configuration roles webadmin authorizations> cd ../../..
caji:configuration> users
caji:configuration users> netuser brendan
caji:configuration users> show
Users:
NAME               USERNAME   UID  TYPE
Brendan Gregg      brendan     130948  Dir
Super-User         root        0    Loc
caji:configuration users> select brendan
caji:configuration users brendan> show
Properties:
    logname = brendan
    fullname = Brendan Gregg
    initial_password = *************
    require_annotation = false
    roles = basic
    kiosk_mode = false
    kiosk_screen = status/dashboard

Children:
    exceptions => Configure this user's exceptions
    preferences => Configure user preferences

caji:configuration users brendan> set roles=basic,webadmin
roles = basic,webadmin (uncommitted)
caji:configuration users brendan> commit
```

▼ Adding Authorizations to a Role using the CLI

1. Go to `configuration roles`.
2. **Type** `select` **followed by the role name**.

3. **Type** `authorizations`.

4. **Type** `create` **to add an authorization**.

5. **Type** `set` `scope=` **followed by the scope name**. Use tab-completion to see the list.

6. **Type** `show` **to see both available filters and authorizations**.

7. **Type** `set` **to set the desired authorizations to true**, and **set the filters (if available)**. Tab-completion helps show which filter settings are valid.

8. **Type** `commit`. The authorization has now been added.

**Example 4** CLI Example for adding the Authorization to Restart the HTTP Service

This example adds the authorization to restart the HTTP service. This example also shows the output of tab-completion, which lists valid input and is useful when determining what are valid scopes and filter options.

```bash
caji:configuration roles> select webadmin
caji:configuration roles webadmin> authorizations
caji:configuration roles webadmin authorizations> create
caji:configuration roles webadmin auth (uncommitted)> set scope=
ad           cluster      net          schema       update
alert        hardware     replication  stat         user
appliance    nas          role         svc          worksheet
caji:configuration roles webadmin auth (uncommitted)> set scope=svc
scope = svc
caji:configuration roles webadmin auth (uncommitted)> show
Properties:
  scope = svc
  service = *
  allow_administer = false
  allow_configure = false
  allow_restart = false

caji:configuration roles webadmin auth (uncommitted)> set service=*tab
  ftp           ipmp      nis           ssh
  ad            http      iscsi       ntp          tags
  smb           identity  ldap        routing      vscan
datalink:igb0  idmap     ndmp         scrk
  dns           interface:igb0  nfs          snmp

caji:configuration roles webadmin auth (uncommitted)> set service=http
  service = http (uncommitted)

caji:configuration roles webadmin auth (uncommitted)> set allow_restart=true
  allow_restart = true (uncommitted)

caji:configuration roles webadmin auth (uncommitted)> commit
```
Deleting Authorizations from a Role using the CLI

1. Go to `configuration roles`.
2. Type `select` followed by the role name.
3. Type `authorizations`.
4. Type `show` to list authorizations.
5. Type `destroy` followed by the authorization name (eg, "auth-001"). The authorization has now been destroyed.

Setting Oracle ZFS Storage Appliance Preferences

This section contains preference settings for your locality, session properties, and SSH keys.

To configure your preferences, use the following sections:
- “Preference Properties” on page 132
- “Setting Preferences using the CLI” on page 133
- “Setting SSH Public Keys using the CLI” on page 134

Preference Properties

When logged into the BUI, you can set the following preferences for your account, but you cannot set other user account preferences.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial login screen</td>
<td>First page the BUI will load after a successful login. By default, this is the Status Dashboard.</td>
</tr>
<tr>
<td>Locality</td>
<td>C by default. C and POSIX Localities support only ASCII characters or plain text. ISO 8859-1 supports the following languages: Afrikaans, Basque, Catalan, Danish, Dutch, English, Faeroese, Finnish, French,</td>
</tr>
</tbody>
</table>
Setting Preferences using the CLI

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Galician, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish and Swedish.</td>
<td></td>
</tr>
<tr>
<td>Session timeout</td>
<td>Time after navigating away from the BUI that the browser will automatically logout the session</td>
</tr>
<tr>
<td>Current session annotation</td>
<td>Annotation text added to audit logs</td>
</tr>
<tr>
<td>Advanced analytics statistics</td>
<td>This will make available additional statistics in Analytics</td>
</tr>
<tr>
<td>SSH Public Keys</td>
<td>RSA/DSA public keys. Text comments can be associated with the keys to help administrators track why they were added. In the BUI, these keys apply only for the current user; to add keys for other users, use the CLI.</td>
</tr>
</tbody>
</table>

**Setting Preferences using the CLI**

- To set user preferences, use the `configuration users` CLI command.

**Example 5** CLI Example for Enabling Advanced Analytics

The following example shows enabling advanced analytics for the "brendan" user account.

```
caji:> configuration users
caji:configuration users> select brendan
caji:configuration users brendan> preferences
caji:configuration users brendan preferences> show
Properties:
locale = C
login_screen = status/dashboard
session_timeout = 15
advanced_analytics = false

Children:
keys => Manage SSH public keys
```

caji:configuration users brendan preferences> set advanced_analytics=true
advanced_analytics = true (uncommitted)
caji:configuration users brendan preferences> commit

**Example 6** CLI Example for Setting a Session Annotation

Set your own preferences in the CLI under `configuration preferences`. The following example shows setting a session annotation for your own account.

```
twofish:> configuration preferences
twofish:configuration preferences> show
Properties:
locale = C
```
Setting SSH Public Keys using the CLI

SSH Public Keys may be needed when automating the execution of CLI scripts from another host. The key text is just the key text itself (usually hundreds of characters), without spaces.

To add an SSH key, use the following CLI commands.

```
caji:> configuration preferences keys
caji:configuration preferences keys> create
caji:configuration preferences key (uncommitted)> set type=DSA
caji:configuration preferences key (uncommitted)> set key="...DSA key text...
key = ...DSA key text...== (uncommitted)
caji:configuration preferences key (uncommitted)> set comment="fw-log1"
comment = fw-log1 (uncommitted)
caji:configuration preferences key (uncommitted)> commit
caji:configuration preferences keys> show
Keys:
NAME     MODIFIED              TYPE   COMMENT
key-000  10/12/2009 10:54:58   DSA    fw-log1
```

Configuring Alerts

This section describes system Alerts, how they are customized, and where to find alert logs. To monitor statistics from Analytics, create custom threshold alerts. To configure the system to respond to certain types of alerts, use Alert actions.

To configure alerts, use the following sections:

- “Alert Categories” on page 135
- “Threshold Alerts” on page 135
- “Adding a Threshold Alert using the BUI” on page 136
- “Adding an Alert Action using the BUI” on page 137
Alert Categories

Important appliance events trigger alerts, which includes hardware and software faults. These alerts appear in the Logs and may also be configured to execute any of the Alert actions.

Alerts are grouped into the following categories:

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster</td>
<td>Cluster events, including link failures and peer errors</td>
</tr>
<tr>
<td>Custom</td>
<td>Events generated from the custom alert configuration</td>
</tr>
<tr>
<td>Hardware Events</td>
<td>Appliance boot and hardware configuration changes</td>
</tr>
<tr>
<td>Hardware Faults</td>
<td>Any hardware fault</td>
</tr>
<tr>
<td>NDMP operations</td>
<td>NDMP TAR/DUMP backup and restore start and finish events. This group is available as “NDMP: backup only” and “NDMP: restore only”</td>
</tr>
<tr>
<td>Network</td>
<td>Network port, datalink, and IP interface events and failures</td>
</tr>
<tr>
<td>Phone Home</td>
<td>Support bundle upload events</td>
</tr>
<tr>
<td>Remote replication</td>
<td>Send and receive events and failures. This group is available as “Remote replication: source only” and “Remote replication: target only”, for just source or target events</td>
</tr>
<tr>
<td>Service failures</td>
<td>Software services failure events</td>
</tr>
<tr>
<td>Thresholds</td>
<td>Custom alerts based on Analytics statistics</td>
</tr>
<tr>
<td>ZFS pool</td>
<td>Storage pool events, including scrub and hot space activation</td>
</tr>
</tbody>
</table>

Threshold Alerts

These are alerts based on the statistics from Analytics. The following are properties when creating threshold alerts:
Adding a Threshold Alert using the BUI

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold</td>
<td>The threshold statistic is from Analytics, and is self descriptive (e.g., &quot;Protocol: NFSv4 operations per second&quot;)</td>
</tr>
<tr>
<td>exceeds/falls below</td>
<td>defines how the threshold value is compared to the current statistic</td>
</tr>
<tr>
<td>Timing: for at least</td>
<td>Duration which the current statistic value must exceed/fall below the threshold</td>
</tr>
<tr>
<td>only between/only during</td>
<td>These properties may be set so that the threshold is only sent during certain times of day - such as business hours</td>
</tr>
<tr>
<td>Repost alert every ... this condition persists.</td>
<td>If enabled, this will re-execute the alert action (such as sending email) every set interval while the threshold breech exists</td>
</tr>
<tr>
<td>Also post alert when this condition clears for at least ...</td>
<td>Send a followup alert if the threshold breech clears for at least the set interval</td>
</tr>
</tbody>
</table>

The “Add Threshold Alert” dialog has been organized so that it can be read as though it is a paragraph describing the alert. The default reads:

Threshold CPU: percent utilization exceeds 95 percent

Timing for at least 5 minutes only between 0:00 and 0:00 only during weekdays

Repost alert every 5 minutes while this condition persists.

Also post alert when this condition clears for at least 5 minutes

▼ Adding a Threshold Alert using the BUI

1. Click the add icon next to "Threshold alerts".

2. Pick the statistic to monitor. You can use "Statistics" to view the statistic to check if it is suitable.

3. Pick exceeds/falls below, and the desired value.

4. Enter the Timing details. The defaults will post the alert only if the threshold has been breached for at least 5 minutes, will repost every 5 minutes, and post after the threshold has cleared for 5 minutes.

5. Select the Alert action from the drop down menu, and fill out the required fields on the right.
6. If desired, continue to add Alert actions by clicking the add icon next to "Alert actions".

7. Click "APPLY" at the top of the dialog.

Adding an Alert Action using the BUI

1. Click the add icon next to "Alert actions".

2. Select the Category, or pick "All events" for everything.

3. Either pick All Events, or a Subset of Events. If the subset is selected, customize the checkbox list to match the desired alerts events.

4. Use the drop down menu in "Alert actions" to select which alert type.

5. Enter details for the Alert action. The "TEST" button can be clicked to create a test alert and execute this alert action (useful for checking if email or SNMP is configured correctly).

6. The add icon next to "Alert actions" can be clicked to add multiple alerts actions.

7. Click "ADD" at the top right.

Adding a Threshold Alert using the CLI

1. Enter the configuration alerts thresholds context, and enter the create command.

2. Enter set statname=where [name is the desired statistic to monitor. To determine the CLI name, enter set statname= and press Tab. For details on each statistic, see "Statistics" and click on the statistic names.

3. Enter set limit=where [number is the desired threshold.

4. Enter commit. Make note of the "watch" identifier, the threshold ID, if you want to later add an alert action for this threshold alert.

5. Enter list to determine the name, including number, of the new threshold alert. Look for a threshold with the same limit and statistic name that you just set.
6. Enter select threshold-where \[number\] is the same number identified in the previous step.

7. Enter list. If necessary, correct any arguments now. By default, the minimum post, frequency, and minimum cleared arguments are set to 5 minutes. This means an alert is posted only if the threshold has been breached for at least 5 minutes, reposts every 5 minutes, and posts after the threshold has cleared for 5 minutes.

8. Enter done, and then enter done again.

\[\textbf{\textit{Adding an Alert Action using the CLI}}\]

1. Enter the configuration alerts actions context, and enter the create command.

2. Go to the "category" property by entering get category = (unset).

3. Enter set category=thresholds.

4. Enter set thresholdid=where \[id\] is the identifier that was automatically created for the threshold alert.

5. Enter commit.

6. Enter list to determine the name, including number, of the new alert action. Look for a threshold without an assigned action and handler.

7. Enter select actions-where \[number\] is the same number identified in the previous step.

8. Enter action, and then enter get.

9. By default, the alert type is email. If this is what you want, skip to the next step. If not, enter set handler=where \[type\] is either snmptrap, syslog, resumedataset, suspenddataset, resumedataset, suspenddataset, or executeworkflow. Then enter get to view the needed arguments. Only snmptrap and syslog do not have arguments.

10. Set each needed argument. For example, to set a subject line for an email alert, enter set subject=where \[subject\] is the desired email subject line.

11. Use the show command to ensure all arguments have been entered.

12. Enter commit, and then enter list. If necessary, correct any arguments now.
13. Enter done, and then enter done again.

**Sending Email Alerts**

An email containing the alert details can be sent. The configuration requires an email address and email subject line. The following example shows an email threshold alert. Details on how the appliance sends mail can be configured on the SMTP service screen.

From aknobody@caji.com Mon Oct 13 15:24:47 2009
Date: Mon, 13 Oct 2009 15:24:21 +0000 (GMT)
From: Appliance on caji <noreply@caji.com>
Subject: High CPU on caji
To: admin@hostname.com

SUNW-MSG-ID: AK-8000-TT, TYPE: Alert, VER: 1, SEVERITY: Minor
EVENT-TIME: Mon Oct 13 15:24:12 2009
PLATFORM: i86pc, CSN: 0809QU085, HOSTNAME: caji
SOURCE: svc:/appliance/kit/akd:default, REV: 1.0
EVENT-ID: 15a53214-c4e7-eae4-da6-a652a51ea29b
DESC: cpu.utilization threshold of 90 is violated.
AUTO-RESPONSE: None.
IMPACT: The impact depends on what statistic is being monitored.
REC-ACTION: The suggested action depends on what statistic is being monitored.
SEE: https://192.168.2.80:215/#maintenance/alert=15a53214-c4e7-eae4-da6-a652a51ea29b

**Sending an SNMP Trap**

An SNMP trap containing alert details can be sent, if an SNMP trap destination is configured in the SNMP service, and that service is online. The following example sends an SNMP trap, as seen from the Net-SNMP tool snmptrapd -P.

`# /usr/sfw/sbin/snmptrapd -P
2009-10-13 15:31:15 NET-SNMP version 5.0.9 Started.
2009-10-13 15:31:34 caji.com [192.168.2.80]:
  iso.3.6.1.2.1.1.3.0 = Timeticks: (2132104431) 246 days, 18:30:44.31
  iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.42.2.225.1.3.0.1
  iso.3.6.1.4.1.42.2.225.1.2.1.2.36.55.99.102.48.97.99.100.52.45.51.48.
  99.49.45.52.99.49.57.45.101.57.99.98.45.97.99.50.55.102.55.49.50.54.
  98.55.57 = STRING: "7cf0acd4-30c1-4c19-e9cb-ac27f7126b79"
  iso.3.6.1.4.1.42.2.225.1.2.1.3.36.55.99.102.48.97.99.100.52.45.51.48.
  99.49.45.52.99.49.57.45.101.57.99.98.45.97.99.50.55.102.55.49.50.54.
  98.55.57 = STRING: "alert.ak.xmlrpc.threshold.violated"
  iso.3.6.1.4.1.42.2.225.1.2.1.4.36.55.99.102.48.97.99.100.52.45.51.48.
  48.99.49.45.52.99.49.57.45.101.57.99.98.45.97.99.50.55.102.55.49.50.
  54.98.55.57 = STRING: "cpu.utilization threshold of 90 is violated."`
Sending Syslog Messages

A syslog message containing alert details can be sent to one or more remote systems, if the Syslog service is enabled. Refer to “Syslog Relay service” on page 253 for example syslog payloads and a description of how to configure syslog receivers on other operating systems.

Resuming/Suspending Datasets

Analytics datasets may be resumed or suspended. This is particularly useful when tracking down sporadic performance issues, and when enabling these datasets 24x7 is not desirable.

For example, imagine you noticed a spike in CPU activity once or twice a week, and other analytics showed an associated drop in NFS performance. You enable some additional datasets, but you don't quite have enough information to prove what the problem is. If you could enable the NFS by hostname and filename datasets, you are certain you will understand the cause a lot better. However those particular datasets can be heavy handed - leaving them enabled 24x7 will degrade performance for everyone. This is where the resume/suspend dataset actions may be of use. A threshold alert could be configured to *resume* paused NFS by hostname and filename datasets, only when the CPU activity spike is detected; a second alert can be configured to then *suspend* those datasets, after a short interval of data is collected. The end result - you collect the data you need only during the issue, and minimize the performance impact of this data collection.

For more information on datasets, see “Datasets” in “Oracle ZFS Storage Appliance Analytics Guide”.

Resuming/Suspending Worksheets

These actions are to resume or suspend an entire Analytics worksheet, which may contain numerous datasets. The reasons for doing this are similar to those for resuming and suspending datasets. For more information, see “Open Worksheets” in “Oracle ZFS Storage Appliance Analytics Guide”.

Executing a Workflow

Workflows may be optionally executed as alert actions. To allow a workflow to be eligible as an alert action, its alert action must be set to *true*. Refer to “Working with Maintenance Workflows” on page 261 for details.
Configuring Clustering

The Oracle ZFS Storage Appliance supports cooperative clustering of appliances. This strategy can be part of an integrated approach to availability enhancement that may also include client-side load balancing, proper site planning, proactive and reactive maintenance and repair, and the single-appliance hardware redundancy built into all Oracle ZFS Storage Appliances.

The clustering feature relies on shared access to storage resources. To configure clustering, both heads must be the same model. Note that the 7420 (with 2Ghz or 2.40GHz CPUs) is based on the same platform and can be clustered with the 7420 (with 1.86GHz or 2.00GHz CPUs).

To configure clustering, use the following sections:
- “Understanding Clustering” on page 141
- “Configuring Clustering using the BUI” on page 162
- “Unconfiguring Clustering using the BUI” on page 164
- “Shutting Down a Clustered Configuration using the CLI” on page 165
- “Shutdown the Stand-by Head using the CLI” on page 166
- “Unconfiguring Clustering using the CLI” on page 166
- “Cabling ZS3-4 and 7x20 Clusters” on page 169
- “Cabling Storage Shelves for Clustering” on page 169

Understanding Clustering

The clustering subsystem incorporated into the series consists of three main building blocks (see Illustration 1). The cluster I/O subsystem and the hardware device provide a transport for inter-head communication within the cluster and are responsible for monitoring the peer's state. This transport is used by the resource manager, which allows data service providers and other management subsystems to interface with the clustering system. Finally, the cluster management user interfaces provide the setup task, resource allocation and assignment, monitoring, and takeover and failback operations. Each of these building blocks is described in detail in the following sections.
FIGURE 19 Clustering Subsystem

Cluster Advantages and Disadvantages

It is important to understand the scope of the Oracle ZFS Storage Appliance clustering implementation. The term 'cluster' is used in the industry to refer to many different technologies with a variety of purposes. We use it here to mean a metasystem comprised of two appliance heads and shared storage, used to provide improved availability in the case in which one of the heads succumbs to certain hardware or software failures. A cluster contains exactly two appliances or storage controllers, referred to for brevity throughout this document as *heads*. 
Each head may be assigned a collection of storage, networking, and other resources from the set available to the cluster, which allows the construction of either of two major topologies. Many people use the terms active-active to describe a cluster in which there are two (or more) storage pools, one of which is assigned to each head along with network resources used by clients to reach the data stored in that pool, and active-passive to refer to which a single storage pool is assigned to the head designated as active along with its associated network interfaces. Both topologies are supported by the Oracle ZFS Storage Appliance. The distinction between these is artificial; there is no software or hardware difference between them and one can switch at will simply by adding or destroying a storage pool. In both cases, if a head fails, the other (its peer) will take control of all known resources and provide the services associated with those resources.

As an alternative to incurring hours or days of downtime while the head is repaired, clustering allows a peer appliance to provide service while repair or replacement is performed. In addition, clusters support rolling upgrade of software, which can reduce the business disruption associated with migrating to newer software. Some clustering technologies have certain additional capabilities beyond availability enhancement; the Oracle ZFS Storage Appliance clustering subsystem was not designed to provide these. In particular, it does not provide for load balancing among multiple heads, improve availability in the face of storage failure, offer clients a unified filesystem namespace across multiple appliances, or divide service responsibility across a wide geographic area for disaster recovery purposes. These functions are likewise outside the scope of this document; however, the Oracle ZFS Storage Appliance and the data protocols it offers support numerous other features and strategies that can improve availability:

- Replication of data, which can be used for disaster recovery at one or more geographically remote sites
- Client-side mirroring of data, which can be done using redundant iSCSI LUNs provided by multiple arbitrarily located storage servers
- Load balancing, which is built into the NFS protocol and can be provided for some other protocols by external hardware or software (applies to read-only data)
- Redundant hardware components including power supplies, network devices, and storage controllers
- Fault management software that can identify failed components, remove them from service, and guide technicians to repair or replace the correct hardware
- Network fabric redundancy provided by LACP and IPMP functionality
- Redundant storage devices (RAID)

Additional information about other availability features can be found in the appropriate sections of this document.

When deciding between a clustered and standalone Oracle ZFS Storage Appliance configuration, it is important to weigh the costs and benefits of clustered operation. It is common practice throughout the IT industry to view clustering as an automatic architectural decision, but this thinking reflects an idealized view of clustering's risks and rewards promulgated by some vendors in this space. In addition to the obvious higher up-front and ongoing hardware and support costs associated with the second head, clustering also imposes
additional technical and operational risks. Some of these risks can be mitigated by ensuring that all personnel are thoroughly trained in cluster operations; others are intrinsic to the concept of clustered operation. Such risks include:

- The potential for application intolerance of protocol-dependent behaviors during takeover,
- The possibility that the cluster software itself will fail or induce a failure in another subsystem that would not have occurred in standalone operation,
- Increased management complexity and a higher likelihood of operator error when performing management tasks,
- The possibility that multiple failures or a severe operator error will induce data loss or corruption that would not have occurred in a standalone configuration, and
- Increased difficulty of recovering from unanticipated software and/or hardware states.

These costs and risks are fundamental, apply in one form or another to all clustered or cluster-capable products on the market (including the Oracle ZFS Storage Appliance), and cannot be entirely eliminated or mitigated. Storage architects must weigh them against the primary benefit of clustering: the opportunity to reduce periods of unavailability from hours or days to minutes or less in the rare event of catastrophic hardware or software failure. Whether that cost/benefit analysis will favor the use of clustering in an Oracle ZFS Storage Appliance deployment will depend on local factors such as SLA terms, available support personnel and their qualifications, budget constraints, the perceived likelihood of various possible failures, and the appropriateness of alternative strategies for enhancing availability. These factors are highly site-, application-, and business-dependent and must be assessed on a case-by-case basis. Understanding the material in the rest of this section will help you make appropriate choices during the design and implementation of your unified storage infrastructure.

**Cluster Terminology**

The terms defined here are used throughout the document. In most cases, they are explained in greater context and detail along with the broader concepts involved. The cluster states and resource types are described in the next section. Refer back to this section for reference as needed.

- export: the process of making a resource inactive on a particular head
- failback: the process of moving from AKCS_OWNER state to AKCS_CLUSTERED, in which all foreign resources (those assigned to the peer) are exported, then imported by the peer
- import: the process of making a resource active on a particular head
- peer: the other appliance in a cluster
- rejoin: to retrieve and resynchronize the resource map from the peer
- resource: a physical or virtual object present, and possibly active, on one or both heads
- takeover: the process of moving from AKCS_CLUSTERED or AKCS_STRIPED state to AKCS_OWNER, in which all resources are imported
Cluster Interconnect I/O

All inter-head communication consists of one or more messages transmitted over one of the three cluster I/O links provided by the CLUSTERON hardware (see illustration below). This device offers two low-speed serial links and one Ethernet link. The use of serial links allows for greater reliability; Ethernet links may not be serviced quickly enough by a system under extremely heavy load. False failure detection and unwanted takeover are the worst way for a clustered system to respond to load; during takeover, requests will not be serviced and will instead be enqueued by clients, leading to a flood of delayed requests after takeover in addition to already heavy load. The serial links used by the Oracle ZFS Storage Appliances are not susceptible to this failure mode. The Ethernet link provides a higher-performance transport for non-heartbeat messages such as rejoin synchronization and provides a backup heartbeat.

All three links are formed using ordinary straight-through EIA/TIA-568B (8-wire, Gigabit Ethernet) cables. To allow for the use of straight-through cables between two identical controllers, the cables must be used to connect opposing sockets on the two connectors as shown below in the section on cabling.

FIGURE 20 ZS3-2 Controller Cluster I/O Ports

<table>
<thead>
<tr>
<th>TABLE 44 ZS3-2 Controller Cluster I/O Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Figure Legend</strong></td>
</tr>
<tr>
<td>1 Serial 0</td>
</tr>
<tr>
<td>5 Serial 1</td>
</tr>
</tbody>
</table>
Clustered heads only communicate with each other over the secure private network established by the cluster interconnects, and never over network interfaces intended for service or administration. Messages fall into two general categories: regular heartbeats used to detect the failure of a remote head, and higher-level traffic associated with the resource manager and the cluster management subsystem. Heartbeats are sent, and expected, on all three links; they are transmitted continuously at fixed intervals and are never acknowledged or retransmitted as all heartbeats are identical and contain no unique information. Other traffic may be sent over any link, normally the fastest available at the time of transmission, and this traffic is acknowledged, verified, and retransmitted as required to maintain a reliable transport for higher-level software.

Regardless of its type or origin, every message is sent as a single 128-byte packet and contains a data payload of 1 to 68 bytes and a 20-byte verification hash to ensure data integrity. The serial
Configuring Clustering

Configuring the Oracle ZFS Storage Appliance

Links run at 115200 bps with 9 data bits and a single start and stop bit; the Ethernet link runs at 1Gbps. Therefore the effective message latency on the serial links is approximately 12.2ms. Ethernet latency varies greatly; while typical latencies are on the order of microseconds, effective latencies to the appliance management software can be much higher due to system load.

Normally, heartbeat messages are sent by each head on all three cluster I/O links at 50ms intervals. Failure to receive any message is considered link failure after 200ms (serial links) or 500ms (Ethernet links). If all three links have failed, the peer is assumed to have failed; takeover arbitration will be performed. In the case of a panic, the panicking head will transmit a single notification message over each of the serial links; its peer will immediately begin takeover regardless of the state of any other links. Given these characteristics, the clustering subsystem normally can detect that its peer has failed within:

- 550ms, if the peer has stopped responding or lost power, or
- 30ms, if the peer has encountered a fatal software error that triggered an operating system panic.

All of the values described in this section are fixed; as an appliance, the Oracle ZFS Storage Appliance does not offer the ability (nor is there any need) to tune these parameters. They are considered implementation details and are provided here for informational purposes only. They may be changed without notice at any time.

**Note** - To avoid data corruption after a physical re-location of a cluster, verify that all cluster cabling is installed correctly in the new location. For more information, see “Preventing ‘Split-Brain’ Conditions” on page 158.

Cluster Resource Management

The resource manager is responsible for ensuring that the correct set of network interfaces is plumbed up, the correct storage pools are active, and the numerous configuration parameters remain in sync between two clustered heads. Most of this subsystem's activities are invisible to administrators; however, one important aspect is exposed. Resources are classified into several types that govern when and whether the resource is imported (made active). Note that the definition of active varies by resource class; for example, a network interface belongs to the net class and is active when the interface is brought up. The three most important resource types are singleton, private, and replica.

Replicas are simplest: they are never exposed to administrators and do not appear on the cluster configuration screen (see Illustration 4). Replicas always exist and are always active on both heads. Typically, these resources simply act as containers for service properties that must be synchronized between the two heads.

Like replicas, singleton resources provide synchronization of state; however, singletons are always active on exactly one head. Administrators can choose the head on which each singleton
should normally be active; if that head has failed, its peer will import the singleton. Singletons are the key to clustering's availability characteristics; they are the resources one typically imagines moving from a failed head to its surviving peer and include network interfaces and storage pools. Because a network interface is a collection of IP addresses used by clients to find a known set of storage services, it is critical that each interface be assigned to the same head as the storage pool clients will expect to see when accessing that interface's address(es). In Illustration 4, all of the addresses associated with the PrimaryA interface will always be provided by the head that has imported pool-0, while the addresses associated with PrimaryB will always be provided by the same head as pool-1.

Private resources are known only to the head to which they are assigned, and are never taken over upon failure. This is typically useful only for network interfaces; see the following discussion of specific use cases.

**FIGURE 22** ZS3-2 Clustering Example

Several other resource types exist; these are implementation details that are not exposed to administrators. One such type is the symbiote, which allows one resource to follow another as it is imported and exported. The most important use of this resource type is in representing the disks and flash devices in the storage pool. These resources are known as disksets and must always be imported before the ZFS pool they contain. Each diskset consists of half the disks in an external storage enclosure; a clustered storage system may have any number of disksets attached (depending on hardware support), and each ZFS pool is formed from the storage devices in one or more disksets. Because disksets may contain ATA devices, they must be explicitly imported and exported to avoid certain affiliation-related behaviors specific to ATA devices used in multipathed environments. Representing disks as resources provides a simple way to perform these activities at the right time. When an administrator sets or changes the ownership of a storage pool, the ownership assignment of the disksets associated with it is transparently changed at the same time. Like all symbiotes, diskset resources do not appear in the cluster configuration user interface.
TABLE 46  Cluster Resource Management

<table>
<thead>
<tr>
<th>Resource</th>
<th>icon</th>
<th>Omnipresent</th>
<th>Taken over on failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINGLETON</td>
<td></td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>REPLICA</td>
<td>None</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>PRIVATE</td>
<td></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SYMBIOTE</td>
<td>None</td>
<td>Same as parent type</td>
<td>Same as parent type</td>
</tr>
</tbody>
</table>

When a new resource is created, it is initially assigned to the head on which it is being created. This ownership cannot be changed unless that head is in the AKCS_OWNER state; it is therefore necessary either to create resources on the head which should own them normally or to take over before changing resource ownership. It is generally possible to destroy resources from either head, although destroying storage pools that are exported is not possible. Best results will usually be obtained by destroying resources on the head which currently controls them, regardless of which head is the assigned owner.

Most configuration settings, including service properties, users, roles, identity mapping rules, SMB autohome rules, and iSCSI initiator definitions are replicated on both heads automatically. Therefore it is never necessary to configure these settings on both heads, regardless of the cluster state. If one appliance is down when the configuration change is made, it will be replicated to the other when it rejoins the cluster on next boot, prior to providing any service. There are a small number of exceptions:

- Share and LUN definitions and options may be set only on the head which has control of the underlying pool, regardless of the head to which that pool is ordinarily assigned.
- The “Identity” service's configuration (i.e., the appliance name and location) is not replicated.
- Names given to chassis are visible only on the head on which they were assigned.
- Each network route is bound to a specific interface. If each head is assigned an interface with an address in a particular subnet, and that subnet contains a router to which the appliances should direct traffic, a route must be created for each such interface, even if the same gateway address is used. This allows each route to become active individually as control of the underlying network resources shifts between the two heads. See Networking Considerations for more details.
- SSH host keys are not replicated and are never shared. Therefore if no private administrative interface has been configured, you may expect key mismatches when attempting to log into the CLI using an address assigned to a node that has failed. The same limitations apply to the SSL certificates used to access the BUI.

The basic model, then, is that common configuration is transparently replicated, and administrators will assign a collection of resources to each appliance head. Those resource assignments in turn form the binding of network addresses to storage resources that clients expect to see. Regardless of which appliance controls the collection of resources, clients are able to access the storage they require at the network locations they expect.
Cluster Takeover and Failback

Clustered head nodes are in one of a small set of states at any given time:

### TABLE 47  Cluster States

<table>
<thead>
<tr>
<th>State</th>
<th>Icon</th>
<th>CLI/BUI Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNCONFIGURED</td>
<td>![Icon]</td>
<td>Clustering is not configured</td>
<td>A system that has no clustering at all is in this state. The system is either being set up or the cluster setup task has never been completed.</td>
</tr>
<tr>
<td>OWNER</td>
<td>![Icon]</td>
<td>Active (takeover completed)</td>
<td>Clustering is configured, and this node has taken control of all shared resources in the cluster. A system enters this state immediately after cluster setup is completed from its user interface, and when it detects that its peer has failed (i.e. after a takeover). It remains in this state until an administrator manually executes a fail-back operation.</td>
</tr>
<tr>
<td>STRIPPED</td>
<td>![Icon]</td>
<td>Ready (waiting for failback)</td>
<td>Clustering is configured, and this node does not control any shared resources. A system is STRIPPED immediately after cluster setup is completed from the user interface of the other node, or following a reboot, power disconnect, or other failure. A node remains in this state until an administrator manually executes a fail-back operation.</td>
</tr>
<tr>
<td>CLUSTERED</td>
<td>![Icon]</td>
<td>Active</td>
<td>Clustering is configured, and both nodes own shared resources according to their resource assignments. If each node owns a ZFS pool and is in the CLUSTERED state, then the two nodes form what is commonly called an active-active cluster.</td>
</tr>
</tbody>
</table>
| -             | ![Icon] | Rejoining cluster ...       | The appliance has recently rebooted, or the appliance

Transitions among these states take place as part of two operations: takeover and failback.

Takeover can occur at any time; as discussed above, takeover is attempted whenever peer failure is detected. It can also be triggered manually using the cluster configuration CLI or BUI. This is useful for testing purposes as well as to perform rolling software upgrades (upgrades in which one head is upgraded while the other provides service running the older software, then the second head is upgraded once the new software is validated). Finally, takeover will occur when a head boots and detects that its peer is absent. This allows service to resume normally when one head has failed permanently or when both heads have temporarily lost power.

Failback never occurs automatically. When a failed head is repaired and booted, it will rejoin the cluster (resynchronizing its view of all resources, their properties, and their ownership) and proceed to wait for an administrator to perform a failback operation. Until then, the original surviving head will continue to provide all services. This allows for a full investigation of the problem that originally triggered the takeover, validation of a new software revision, or other administrative tasks prior to the head returning to production service. Because failback is disruptive to clients, it should be scheduled according to business-specific needs and processes. There is one exception: Suppose that head A has failed and head B has taken over. When head A rejoins the cluster, it becomes eligible to take over if it detects that head B is absent or has failed. The principle is that it is always better to provide service than not, even if there has not yet been an opportunity to investigate the original problem. So while failback to a previously-failed head will never occur automatically, it may still perform takeover at any time.

When you set up a cluster, the initial state consists of the node that initiated the setup in the OWNER state and the other node in the STRIPPED state. After performing an initial failback operation to hand the STRIPPED node its portion of the shared resources, both nodes are CLUSTERED. If both cluster nodes fail or are powered off, then upon simultaneous startup they will arbitrate and one of them will become the OWNER and the other STRIPPED.

During failback all foreign resources (those assigned to the peer) are exported, then imported by the peer. A pool that cannot be imported because it is faulted will trigger reboot of the STRIPPED node. An attempt to failback with a faulted pool can reboot the STRIPPED node as a result of the import failure.
Configuration Changes in a Clustered Environment

The vast majority of appliance configuration is represented as either service properties or share/LUN properties. While share and LUN properties are stored with the user data on the storage pool itself (and thus are always accessible to the current owner of that storage resource), service configuration is stored within each head. To ensure that both heads provide coherent service, all service properties must be synchronized when a change occurs or a head that was previously down rejoins with its peer. Since all services are represented by replica resources, this synchronization is performed automatically by the appliance software any time a property is changed on either head.

It is therefore unnecessary and redundant for administrators to replicate configuration changes. Standard operating procedures should reflect this attribute and call for making changes to only one of the two heads once initial cluster configuration has been completed. Note as well that the process of initial cluster configuration will replicate all existing configuration onto the newly-configured peer. Generally, then, we derive two best practices for clustered configuration changes:

- Make all storage- and network-related configuration changes on the head that currently controls (or will control, if a new resource is being created) the underlying storage or network interface resources.
- Make all other changes on either head, but not both. Site policy should specify which head is to be considered the master for this purpose, and should in turn depend on which of the heads is functioning and the number of storage pools that have been configured. Note that the appliance software does not make this distinction.

The problem of amnesia, in which disjoint configuration changes are made and subsequently lost on each head while its peer is not functioning, is largely overstated. This is especially true of the Oracle ZFS Storage Appliance, in which no mechanism exists for making independent changes to system configuration on each head. This simplification largely alleviates the need for centralized configuration repositories and argues for a simpler approach: whichever head is currently operating is assumed to have the correct configuration, and its peer will be synchronized to it when booting. While future product enhancements may allow for selection of an alternate policy for resolving configuration divergence, this basic approach offers simplicity and ease of understanding: the second head will adopt a set of configuration parameters that are already in use by an existing production system (and are therefore highly likely to be correct). To ensure that this remains true, administrators should ensure that a failed head rejoins the cluster as soon as it is repaired.

Clustering Considerations for Storage

When sizing an Oracle ZFS Storage Appliance for use in a cluster, two additional considerations gain importance. Perhaps the most important decision is whether all storage pools will be assigned ownership to the same head, or split between them. There are several trade-offs here, as shown in the table below. Generally, pools should be configured on a single
head except when optimizing for throughput during nominal operation or when failed-over performance is not a consideration. The exact changes in performance characteristics when in the failed-over state will depend to a great deal on the nature and size of the workload(s). Generally, the closer a head is to providing maximum performance on any particular axis, the greater the performance degradation along that axis when the workload is taken over by that head's peer. Of course, in the multiple pool case, this degradation will apply to both workloads.

Note that in either configuration, any ReadZilla devices can be used only when the pool to which they are assigned is imported on the head that has been assigned ownership of that pool. That is, when a pool has been taken over due to head failure, read caching will not be available for that pool even if the head that has imported it also has unused ReadZillas installed. For this reason, ReadZillas in an active-passive cluster should be configured as described in “Storage Configuration” on page 77. This does not apply to LogZilla devices, which are located in the storage fabric and are always accessible to whichever head has imported the pool.

### TABLE 48 Clustering Considerations for Storage

<table>
<thead>
<tr>
<th>Variable</th>
<th>Single Node ownership</th>
<th>Multiple pools owned by different heads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total throughput (nominal operation)</td>
<td>Up to 50% of total CPU resources, 50% of DRAM, and 50% of total network connectivity can be used to provide service at any one time. This is straightforward: only a single head is ever servicing client requests, so the other is idle.</td>
<td>All CPU and DRAM resources can be used to provide service at any one time. Up to 50% of all network connectivity can be used at any one time (dark network devices are required on each head to support failover).</td>
</tr>
<tr>
<td>Total throughput (failed over)</td>
<td>No change in throughput relative to nominal operation.</td>
<td>100% of the surviving head's resources will be used to provide service. Total throughput relative to nominal operation may range from approximately 40% to 100%, depending on utilization during nominal operation.</td>
</tr>
<tr>
<td>I/O latency (failed over)</td>
<td>ReadZilla is not available during failed-over operation, which may significantly increase latencies for read-heavy workloads that fit into available read cache. Latency of write operations is unaffected.</td>
<td>ReadZilla is not available during failed-over operation, which may significantly increase latencies for read-heavy workloads that fit into available read cache. Latency of both read and write operations may be increased due to greater contention for head resources. This is caused by running two workloads on the surviving head instead of the usual one. When nominal workloads on each head approach the head's maximum capabilities, latencies in the failed-over state may be extremely high.</td>
</tr>
<tr>
<td>Storage flexibility</td>
<td>All available physical storage can be used by shares and LUNs.</td>
<td>Only the storage allocated to a particular pool can be used by that pool's shares and LUNs. Storage is not shared across pools, so if one</td>
</tr>
</tbody>
</table>
Clustering Considerations for Networking

Network device, datalink, and interface failures do not cause a clustered subsystem head to fail. To protect against network failures inside or outside of the appliance, IPMP and/or LACP should be used. A comprehensive approach to availability requires the correct configuration of the network and a network-wide plan for redundancy.

FIGURE 23 Clustering for Networking

Network interfaces can be configured as either singleton or private resources, provided they have a static IP configuration. Interfaces configured using DHCP must be private and using DHCP in clusters is discouraged. When configured as a singleton resource, all datalinks and devices used to construct an interface can be active on only one head at a time. Likewise, corresponding devices on each head must be attached to the same networks in order for service to be provided in a failed-over state. An example of this is shown in the previous diagram.
For a cluster to operate correctly when you construct network interfaces from devices and
datalinks, it is essential that each singleton interface has a device using the same identifier and
capabilities available on both heads. Since device identifiers depend on the device type and the
order in which they are first detected by the appliance, clustered heads MUST have identical
hardware installed. Each slot in both heads must be populated with identical hardware and slots
must be populated in the same order on both heads. Your qualified Oracle reseller or service
representative can assist in planning hardware upgrades that meet these requirements.

A route is always bound explicitly to a single network interface. Routes are represented within
the resource manager as symbiotes and can become active only when the interfaces to which
they are bound are operational. Therefore, a route bound to an interface which is currently
in standby mode (exported) has no effect until the interface is activated during the takeover
process. This is important when two pools are configured and are made available to a common
subnet. If a subnet is home to a router that is used by the appliances to reach one or more other
networks, a separate route (for example, a second default route), must be configured and bound
to each of the active and standby interfaces attached to that subnet.

Example:

- Interface e1000g3 is assigned to 'alice' and e1000g4 is assigned to 'bob'.
- Each interface has an address in the 172.16.27.0/24 network and can be used to provide
  service to clients in the 172.16.64.0/22 network, reachable via 172.16.27.1.
- Two routes should be created to 172.16.64.0/22 via 172.16.27.1; one should be bound to
e1000g3 and the other to e1000g4.

It is a good idea to assign each clustered head an IP address used only for administration (most
likely on a dedicated management network) and to designate the interface as a private resource.
This ensures that it is possible to reach a functioning head from the management network even
if it is in an AKCS_STRIPPEP state and awaiting failback. This is important if services such as
LDAP and Active Directory are in use and require access to other network resources when the
head is not providing service. If this is not practical, the service processor should be attached to
a reliable network and/or serial terminal concentrator so that the head can be managed using the
system console.

If neither of these actions is taken, it is impossible to manage or monitor a newly-booted head
until failback is completed. You may want to monitor or manage the head that is providing
service for a particular storage pool. This is likely to be useful when you want to modify
some aspect of the storage itself such as modifying a share property or create a new LUN.
This can be done by using one of the service interfaces to perform administrative tasks or by
allocating a separate singleton interface to be used only for managing the pool to which it is
matched. In either case, the interface should be assigned to the same head as the pool it is used
to manage.

**Private Local IP Interfaces**

Use the following guidelines when creating private local IP interfaces:
Creating an IP interface with the same name as a private IP interface on cluster peer, results in the local creation of a private IP interface.

- Datalinks in use by the peer's private interfaces can not be deleted and the delete button is greyed out.
- IP interfaces that belong to an IPMP group must all be of the same type and belong to the same head. To create an IPMP group you must use either all singleton or all private IP interfaces and your cluster node must be the owner of these interfaces.
- The IPMP group type is set only at creation, and is determined by the type of underlying links.
- IP interfaces that belong to IPMP groups do not appear on the Cluster:Resources page because IP interface ownership cannot be modified independently of the IPMP group ownership.
- Private IPMP groups do not appear in the Cluster:Resources page because this type or ownership cannot be modified.

**Clustering Considerations for Infiniband**

Like a network built on top of ethernet devices, an Infiniband network needs to be part of a redundant fabric topology in order to guard against network failures inside and outside of the appliance. The network topology should include IPMP to protect against network failures at the link level with a broader plan for redundancy for HCAs, switches and subnet managers.

**FIGURE 24** Clustering Considerations for Infiniband

To ensure proper cluster configuration, each head must be populated with identical HCAs in identical slots. Furthermore, each corresponding HCA port must be configured into the same partition (pkey) on the subnet manager with identical membership privileges and attached to the same network. To reduce complexity and ensure proper redundancy, it is recommended that each port belong to only one partition in the Infiniband sub-network. Network interfaces may be configured as either singleton or private resources, provided they have static IP configuration. When configured as a singleton resource, all of the IB partition datalinks and devices used to construct an interface may be active on only one head at any given time. A concrete example of
this is shown in the illustration above. Changes to partition membership for corresponding ports must happen at the same time and in a manner consistent with the clustering rules above. Your qualified Oracle reseller or service representative can assist in planning hardware upgrades that will meet these requirements.

The following illustration shows cluster configuration for subnet manager redundancy. Greater redundancy is achieved by connecting two dual-port HCAs to a redundant pair of server switches.

**FIGURE 25**  Cluster Configuration for Subnet Manager Redundancy
Preventing 'Split-Brain' Conditions

A common failure mode in clustered systems is known as split-brain; in this condition, each of the clustered heads believes its peer has failed and attempts takeover. Absent additional logic, this condition can cause a broad spectrum of unexpected and destructive behavior that can be difficult to diagnose or correct. The canonical trigger for this condition is the failure of the communication medium shared by the heads; in the case of the Oracle ZFS Storage Appliance, this would occur if the cluster I/O links fail. In addition to the built-in triple-link redundancy (only a single link is required to avoid triggering takeover), the appliance software will also perform an arbitration procedure to determine which head should continue with takeover.

A number of arbitration mechanisms are employed by similar products; typically they entail the use of quorum disks (using SCSI reservations) or quorum servers. To support the use of ATA disks without the need for additional hardware, the Oracle ZFS Storage Appliance uses a different approach relying on the storage fabric itself to provide the required mutual exclusivity. The arbitration process consists of attempting to perform a SAS ZONE LOCK command on each of the visible SAS expanders in the storage fabric, in a predefined order. Whichever appliance is successful in its attempts to obtain all such locks will proceed with takeover; the other will reset itself. Since a clustered appliance that boots and detects that its peer is unreachable will attempt takeover and enter the same arbitration process, it will reset in a continuous loop until at least one cluster I/O link is restored. This ensures that the subsequent failure of the other head will not result in an extended outage. These SAS zone locks are released when failback is performed or approximately 10 seconds has elapsed since the head in the AKCS_OWNER state most recently renewed its own access to the storage fabric.

This arbitration mechanism is simple, inexpensive, and requires no additional hardware, but it relies on the clustered appliances both having access to at least one common SAS expander in the storage fabric. Under normal conditions, each appliance has access to all expanders, and arbitration will consist of taking at least two SAS zone locks. It is possible, however, to construct multiple-failure scenarios in which the appliances do not have access to any common expander. For example, if two of the SAS cables are removed or a disk shelf is powered down, each appliance will have access to disjoint subsets of expanders. In this case, each appliance will successfully lock all reachable expanders, conclude that its peer has failed, and attempt to proceed with takeover. This can cause unrecoverable hangs due to disk affiliation conflicts and/or severe data corruption.

Note that while the consequences of this condition are severe, it can arise only in the case of multiple failures (often only in the case of 4 or more failures). The clustering solution embedded in the Oracle ZFS Storage Appliance is designed to ensure that there is no single point of failure, and to protect both data and availability against any plausible failure without adding undue cost or complexity to the system. It is still possible that massive multiple failures will cause loss of service and/or data, in much the same way that no RAID layout can protect against an unlimited number of disk failures.
Fortunately, most such failure scenarios arise from human error and are completely preventable by installing the hardware properly and training staff in cluster setup and management best practices. Administrators should always ensure that all three cluster I/O links are connected and functional (see illustration), and that all storage cabling is connected as shown in the setup poster delivered with your appliances. It is particularly important that two paths are detected to each disk shelf (see illustration) before placing the cluster into production and at all times afterward, with the obvious exception of temporary cabling changes to support capacity increases or replacement of faulty components. Administrators should use alerts to monitor the state of cluster interconnect links and disk shelf paths and correct any failures promptly. Ensuring that proper connectivity is maintained will protect both availability and data integrity if a hardware or software component fails.
Configuring Clustering

Estimating and Reducing Takeover Impact

There is an interval during takeover and failback during which access to storage cannot be provided to clients. The length of this interval varies by configuration, and the exact effects on clients depends on the protocol(s) they are using to access data. Understanding and mitigating these effects can make the difference between a successful cluster deployment and a costly failure at the worst possible time.

NFS (all versions) clients typically hide outages from application software, causing I/O operations to be delayed while a server is unavailable. NFSv2 and NFSv3 are stateless protocols that recover almost immediately upon service restoration. NFSv4 incorporates a client grace period at startup, during which I/O typically cannot be performed. The duration of this grace period can be tuned in the Oracle ZFS Storage Appliance (see illustration); reducing it will reduce the apparent impact of takeover and/or failback. For planned outages, the Oracle ZFS Storage Appliance provides grace-less recovery for NFSv4 clients, which avoids the grace period delay. For more information about grace-less recovery, see the Grace period property in NFS “Properties” on page 183.
iSCSI behavior during service interruptions is initiator-dependent, but initiators will typically recover if service is restored within a client-specific timeout period. Check your initiator’s documentation for additional details. The iSCSI target will typically be able to provide service as soon as takeover is complete, with no additional delays.

SMB, FTP, and HTTP/WebDAV are connection-oriented protocols. Because the session states associated with these services cannot be transferred along with the underlying storage and network connectivity, all clients using one of these protocols will be disconnected during a takeover or failback, and must reconnect after the operation completes.

While several factors affect takeover time (and its close relative, failback time), in most configurations these times will be dominated by the time required to import the diskset resource(s). Typical import times for each diskset range from 15 to 20 seconds, linear in the number of disksets. Recall that a diskset consists of one half of one disk shelf, provided the disk bays in that half-disk shelf have been populated and allocated to a storage pool. Unallocated disks and empty disk bays have no effect on takeover time. The time taken to import diskset resources is unaffected by any parameters that can be tuned or altered by administrators, so administrators planning clustered deployments should either:

- limit installed storage so that clients can tolerate the related takeover times, or
- adjust client-side timeout values above the maximum expected takeover time.

Note that while diskset import usually comprises the bulk of takeover time, it is not the only factor. During the pool import process, any intent log records must be replayed, and each share and LUN must be shared via the appropriate service(s). The amount of time required to perform these activities for a single share or LUN is very small - on the order of tens of milliseconds - but with very large share counts this can contribute significantly to takeover times. Keeping the
number of shares relatively small - a few thousand or fewer - can therefore reduce these times considerably.

Failback time is normally greater than takeover time for any given configuration. This is because failback is a two-step operation: first, the source appliance exports all resources of which it is not the assigned owner, then the target appliance performs the standard takeover procedure on its own assigned resources only. Therefore it will always take longer to failback from head A to head B than it will take for head A to take over from head B in case of failure. This additional failback time is much less dependent upon the number of disksets being exported than is the takeover time, so keeping the number of shares and LUNs small can have a greater impact on failback than on takeover. It is also important to keep in mind that failback is always initiated by an administrator, so the longer service interruption it causes can be scheduled for a time when it will cause the lowest level of business disruption.

Note: Estimated times cited in this section refer to software/firmware version 2009.04.10.1-0. Other versions may perform differently, and actual performance may vary. It is important to test takeover and its exact impact on client applications prior to deploying a clustered appliance in a production environment.

### Configuring Clustering using the BUI

1. **Connect power and at least one Ethernet cable to each appliance.**

2. **Cable together the cluster interconnect controllers as described below under Node Cabling.** You can also proceed with cluster setup and add these cables dynamically during the setup process.

3. **Cable together the HBAs to the shared disk shelf (shelves) as shown in the Disk Shelf Cabling diagrams in the setup poster that came with your appliance.**

4. **Power on both appliances - but do not begin configuration.** Select only one of the two appliances from which you will perform configuration; the choice is arbitrary. This will be referred to as the primary appliance for configuration purposes. Connect to and access the serial console of that appliance, and perform the initial tty-based configuration on it in the same manner as you would when configuring a standalone appliance. Note: Do not perform the initial tty-based configuration on the secondary appliance; it will be automatically configured for you during cluster setup.

5. **On the primary appliance, enter either the BUI or CLI to begin cluster setup.** Cluster setup can be selected as part of initial setup if the cluster interconnect controller has been installed. Alternately, you can perform standalone configuration at this time, deferring cluster setup until later. In the latter case, you can perform the cluster configuration task by clicking the Setup button in Configuration->Cluster.
6. At the first step of cluster setup, you will be shown a diagram of the active cluster links: you should see three solid blue wires on the screen, one for each connection. If you don't, add the missing cables now. Once you see all three wires, you are ready to proceed by clicking the Commit button.

7. Enter the appliance name and initial root password for the second appliance (this is equivalent to performing the initial serial console setup for the new appliance). When you click the Commit button, progress bars will appear as the second appliance is configured.

8. If you are setting up clustering as part of initial setup of the primary appliance, you will now be prompted to perform initial configuration as you would be in the single-appliance case. All configuration changes you make will be propagated automatically to the other appliance. Proceed with initial configuration, taking into consideration the following restrictions and caveats: Network interfaces configured via DHCP cannot be failed over between heads, and therefore cannot be used by clients to access storage. Therefore, be sure to assign static IP addresses to any network interfaces which will be used by clients to access storage. If you selected a DHCP-configured network interface during tty-based initial configuration, and you wish to use that interface for client access, you will need to change its address type to Static before proceeding. Best practices include configuring and assigning a private network interface for administration to each head, which will enable administration via either head over the network (BUI or CLI) regardless of the cluster state. If routes are needed, be sure to create a route on an interface that will be assigned to each head. See the previous section for a specific example.

9. Proceed with initial configuration until you reach the storage pool step. Each storage pool can be taken over, along with the network interfaces clients use to reach that storage pool, by the cluster peer when takeover occurs. If you create two storage pools, each head will normally provide clients with access to the pool assigned to it; if one of the heads fails, the other will provide clients with access to both pools. If you create a single pool, the head which is not assigned a pool will provide service to clients only when its peer has failed. Storage pools are assigned to heads at the time you create them; the storage configuration dialog offers the option of creating a pool assigned to each head independently. The smallest unit of storage that may be assigned to a pool is one disk. If you create multiple pools, there is no requirement that they must be the same size. Note that fewer pools with more disks per pool are preferred because they simplify management and provide a higher percentage of overall usable capacity. It is recommended that each pool includes a minimum of 8 disks, and ideally more, across all disk shelves.

10. After completing basic configuration, you will have an opportunity to assign resources to each head. Typically, you will need to assign only network interfaces; storage pools were automatically assigned during the storage configuration step.
11. Commit the resource assignments and perform the initial fail-back from the Cluster User Interface, described below. If you are still executing initial setup of the primary appliance, this screen will appear as the last in the setup sequence. If you are executing cluster setup manually after an initial setup, go to the Configuration/Cluster screen to perform these tasks. Refer to Cluster User Interface below for the details.

▼ Unconfiguring Clustering using the BUI

Unconfiguring clustering is a destructive operation that returns one of the clustered storage controllers to its factory default configuration and reassigns ownership of all resources to the surviving peer. There are two reasons to unconfiguring clustering. You no longer wish to use clustering; instead, you wish to configure two independent storage appliances. You are replacing a failed storage controller with new hardware or a storage controller with factory-fresh appliance software (typically this replacement is performed by your service provider).

1. Select the storage controller that will be reset to its factory configuration. Note that if replacing a failed storage controller, you can skip to step 3, provided that the failed storage controller will not be returned to service at your site.

2. From the system console of the storage controller that will be reset to its factory configuration, perform a factory reset.

3. The storage controller will reset, and its peer will begin takeover normally. NOTE: Prior to allowing the factory-reset storage controller to begin booting (i.e., prior to progressing beyond the boot menu), power it off and wait for its peer to complete takeover.

4. Detach the cluster interconnect cables (see above) and detach the powered-off storage controller from the cluster's external storage enclosures.

5. On the remaining storage controller, click the Unconfig button on the Configuration -> Clustering screen. All resources will become assigned to that storage controller, and the storage controller will no longer be a member of any cluster.

6. The detached storage controller, if any, can now be attached to its own storage, powered on, and configured normally. If you are replacing a failed storage controller, attach the replacement to the remaining storage controller and storage and begin the cluster setup task described above.
Note - If your cluster had 2 or more pools, ownership of all pools will be assigned to the remaining storage controller after unconfiguration. In software versions prior to 2010.Q1.0.0, this was not a supported configuration; if you are running an older software version, you must do one of: destroy one or both pools, attach a replacement storage controller, perform the cluster setup task described above, and reassign ownership of one of the pools to the replacement storage controller, or upgrade to 2010.Q1.0.0 or a later software release which contains support for multiple pools per storage controller.

Shutting Down a Clustered Configuration using the CLI

1. Verify the cluster state, using the following CLI commands:

   nas-7420-1a:> configuration cluster
   nas-7420-1a:configuration cluster> show

2. The following is an example of the cluster properties: state indicates the status of the head where you ran the command; peer_state indicates the status of the other head.

   state = AKCS_OWNER
   description = Active (takeover completed)
   peer_asn = 365ed33c-3b9d-c533-9349-8014e9da0408
   peer_hostname = nas-7420-1b
   peer_state = AKCS_STRIPPED
   peer_description = Ready (waiting for failback)

3. Use the following table to verify the node status.

<table>
<thead>
<tr>
<th>This Node</th>
<th>Other Node</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKCS_CLUSTERED</td>
<td>AKCS_CLUSTERED</td>
<td>Both nodes are running in normal condition.</td>
</tr>
<tr>
<td>AKCS_OWNER</td>
<td>AKCS_STRIPPED</td>
<td>This node has all the resources and is in active node. The other node is in stand-by and has no resources.</td>
</tr>
<tr>
<td>AKCS_OWNER</td>
<td>rebooting</td>
<td>Another node is rebooting and this node has all resources.</td>
</tr>
<tr>
<td>AKCS_OWNER</td>
<td>unknown</td>
<td>This node does not know the partner.</td>
</tr>
</tbody>
</table>
Shutdown the Stand-by Head using the CLI

1. Shutdown the stand-by head, using the CLI to run the following commands:

   nas-7420-1b:configuration cluster> cd /
   nas-7420-1b:> maintenance system poweroff
   This will turn off power to the appliance. Are you sure? (Y/N)

2. To verify that you want to shut down the other head, type Y.

   Note - If both heads have a status of AKCS_CLUSTERED, a takeover of the surviving head begins automatically.

3. Confirm that the stand-by head is powered off, and the cluster state is OWNER/unknown.

4. Shut down the active head, using the CLI to run the following commands:

   nas-7420-1a:configuration cluster> cd /
   nas-7420-1a:> maintenance system poweroff
   This will turn off power to the appliance. Are you sure? (Y/N)

5. To verify that you want to shut down the active head, type Y.

6. Confirm that both heads are powered off. From the ILOM prompt, run:

   -> show /SYS power_state

7. Power off the disk shelves.

Unconfiguring Clustering using the CLI

Unconfiguring clustering is a destructive operation that returns one of the clustered storage controllers to its factory default configuration and reassigns ownership of all resources to the surviving peer. There are two reasons to unconfiguring clustering. You no longer wish to use clustering; instead, you wish to configure two independent storage appliances. You are
replacing a failed storage controller with new hardware or a storage controller with factory-fresh appliance software (typically this replacement is performed by your service provider).

- **Unconfiguring clustering in the CLI operates the same as the BUI unconfig button. If a user attempts to unconfig a cluster when it is not in a correct state, an error appears.**

```
configuration cluster> help
Subcommands that are valid in this context:

resources           => Configure resources
help [topic]         => Get context-sensitive help. If [topic] is specified, it must be one of "builtins", "commands", "general", "help", "script" or "properties".
show                 => Show information pertinent to the current context
done                 => Finish operating on "cluster"
get [prop]           => Get value for property [prop]. ("help properties" for valid properties.) If [prop] is not specified, returns values for all properties.
setup                => Run through initial cluster setup
failback             => Fail back all resources assigned to the cluster peer
takeover             => Take over all resources assigned to the cluster peer
unconfig             => Unconfigure the cluster
links                => Report the state of the cluster links
```
Cabling ZS3-2 Clusters

FIGURE 29  ZS3-2 Cluster Cabling

The ZS3-2 controller provides three redundant links that enable the heads to communicate: two serial links (the first two connectors) and an Ethernet link (the third connector).

Using straight-through Cat 5-or-better Ethernet cables, (three 1m cables ship with your cluster configuration), connect the head node according to the diagram at left.

The cluster cabling can be performed either prior to powering on either head node, or can be performed live while executing the cluster setup guided task. The user interface will show the status of each link, as shown later in this page. You must have established all three links before cluster configuration will proceed.
Cabling ZS4-4, ZS3-4, and 7x20 Clusters

**FIGURE 30** ZS4-4, ZS3-4, and 7x20 Cluster Cabling

The ZS4-4, ZS3-4, and 7x20 controllers provide three redundant links that enable the heads to communicate: two serial links (the outer two connectors) and an Ethernet link (the middle connector).

Using straight-through Cat 5-or-better Ethernet cables, (three 1m cables ship with your cluster configuration), connect the head node according to the diagram at left.

The cluster cabling can be performed either prior to powering on either head node, or can be performed live while executing the cluster setup guided task. The user interface will show the status of each link, as shown later in this page. You must have established all three links before cluster configuration will proceed.

Cabling Storage Shelves for Clustering

You need to attach your storage shelves to both appliances before beginning cluster configuration. See “Installation” in “Oracle ZFS Storage Appliance Installation Guide” or follow the Quick Setup poster that shipped with your system.

The Configuration > Cluster view provides a graphical overview of the status of the cluster card, the cluster head node states, and all of the resources.
The interface contains the following objects:

- A thumbnail picture of each system, with the system whose administrative interface is being accessed shown at left. Each thumbnail is labeled with the canonical appliance name, and its current cluster state (the icon above, and a descriptive label).
- A thumbnail of each cluster card connection that dynamically updates with the hardware: a solid line connects a link when that link is connected and active, and the line disappears if that connection is broken or while the other system is restarting/rebooting.
- A list of the PRIVATE and SINGLETON resources (see Introduction, above) currently assigned to each system, shown in lists below the thumbnail of each cluster node, along with various attributes of the resources.
- For each resource, the appliance to which that resource is assigned (that is, the appliance that will provide the resource when both are in the CLUSTERED state). When the current appliance is in the OWNER state, the owner field is shown as a pop-up menu that can be edited and then committed by clicking Apply.
- For each resource, a lock icon indicating whether or not the resource is PRIVATE. When the current appliance is in either of the OWNER or CLUSTERED states, a resource can be locked to it (made PRIVATE) or unlocked (made a SINGLETON) by clicking the lock icon and then clicking Apply. Note that PRIVATE resources belonging to the remote peer will not be displayed on either resource list.

The BUI contains the following buttons:

**TABLE 49**  Shelf Cabling Interface Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup</td>
<td>If the cluster is not yet configured, execute the cluster setup guided task, and then return to the current screen. See above for a detailed description of this task.</td>
</tr>
<tr>
<td>Unconfig</td>
<td>Upgrade a node to standalone operation by unconfiguring the cluster. See below for a detailed description of this task.</td>
</tr>
<tr>
<td>Apply</td>
<td>If resource modifications are pending (rows highlighted in yellow), commit those changes to the cluster.</td>
</tr>
<tr>
<td>Revert</td>
<td>If resource modifications are pending (rows highlighted in yellow), revert those changes and show the current cluster configuration.</td>
</tr>
<tr>
<td>Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Failback</td>
<td>If the current appliance (left-hand side) is the OWNER, fail-back resources owned by the other appliance to it, leaving both nodes in the CLUSTERED state (active/active).</td>
</tr>
<tr>
<td>Takeover</td>
<td>If the current appliance (left-hand side) is either CLUSTERED or STRIPPED, force the other appliance to reboot, and take-over its resources, making the current appliance the OWNER.</td>
</tr>
</tbody>
</table>
The Services screen features a side panel for quick navigation between services.

You can configure the following appliance services:

**FIGURE 32** Services Configuration BUI Page

To configure appliance services, use the following sections:

- “Working with NFS” on page 183
Working with Oracle ZFS Storage Appliance Services

The appliance provides the services shown in the following tables.

To work with the appliance services, see the following sections:

- “Data Services” on page 175
- “Required Service Ports” on page 177
- “Services BUI Page Icons and Buttons” on page 177
- “Configuring Services using the CLI” on page 178
- “Viewing a Specific Service Screen using the BUI” on page 179
- “Viewing a Specific Service Screen” on page 180
- “Enabling a Service using the BUI” on page 180
Data Services

TABLE 50   Available Data Services

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
<th>Ports Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>“NFS” on page 183</td>
<td>Filesystem access via the NFSv3 and NFSv4 protocols</td>
<td>111 and 2049</td>
</tr>
<tr>
<td>“iSCSI” on page 187</td>
<td>LUN access via the iSCSI protocol</td>
<td>3260 and 3205</td>
</tr>
<tr>
<td>“SMB” on page 189</td>
<td>Filesystem access via the SMB protocol</td>
<td>SMB-over-NetBIOS 139</td>
</tr>
<tr>
<td>SMB-over-TCP 445</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NetBIOS Datagram 138</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NetBIOS Name Service 137</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“FTP” on page 205</td>
<td>Filesystem access via the FTP protocol</td>
<td>21</td>
</tr>
<tr>
<td>“HTTP” on page 206</td>
<td>Filesystem access via the HTTP protocol</td>
<td>80</td>
</tr>
<tr>
<td>“NDMP” on page 208</td>
<td>NDMP host service</td>
<td>10000</td>
</tr>
<tr>
<td>“Remote Replication” on page 215</td>
<td>Remote replication</td>
<td>216</td>
</tr>
<tr>
<td>“Shadow Migration” on page 216</td>
<td>Shadow data migration</td>
<td></td>
</tr>
<tr>
<td>“SFTP” on page 216</td>
<td>Filesystem access via the SFTP protocol</td>
<td>218</td>
</tr>
<tr>
<td>“SRP” on page 219</td>
<td>Block access via the SRP protocol</td>
<td></td>
</tr>
<tr>
<td>“TFTP” on page 219</td>
<td>Filesystem access via the TFTP protocol</td>
<td></td>
</tr>
<tr>
<td>“Virus Scan” on page 220</td>
<td>Filesystem virus scanning</td>
<td></td>
</tr>
</tbody>
</table>

Note - UIDs and GIDs from 0-99 are reserved by the operating system vendor for use in future applications. Their use by end system users or vendors of layered products is not supported and may cause security related issues with future applications.
### TABLE 51  Available Directory Services

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
<th>Ports Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;NIS&quot; on page 222</td>
<td>Authenticate users and groups from an NIS service</td>
<td></td>
</tr>
<tr>
<td>&quot;LDAP&quot; on page 224</td>
<td>Authenticate users and groups from an LDAP directory</td>
<td>389</td>
</tr>
<tr>
<td>&quot;Active Directory&quot; on page 228</td>
<td>Authenticate users with a Microsoft Active Directory Server</td>
<td></td>
</tr>
<tr>
<td>&quot;Identity Mapping&quot; on page 233</td>
<td>Map between Windows entities and Unix IDs</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 52  Service Settings

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
<th>Ports Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;DNS&quot; on page 239</td>
<td>Domain name service client</td>
<td>53</td>
</tr>
<tr>
<td>&quot;Dynamic Routing&quot; on page 241</td>
<td>RIP and RIPng dynamic routing protocols</td>
<td></td>
</tr>
<tr>
<td>&quot;IPMP&quot; on page 241</td>
<td>IP Multipathing for IP fail-over</td>
<td></td>
</tr>
<tr>
<td>&quot;NTP&quot; on page 242</td>
<td>Network time protocol client</td>
<td></td>
</tr>
<tr>
<td>&quot;Phone Home&quot; on page 245</td>
<td>Product registration and support configuration</td>
<td>443</td>
</tr>
<tr>
<td>&quot;Service Tags&quot; on page 248</td>
<td>Product inventory support</td>
<td>443</td>
</tr>
<tr>
<td>&quot;SMTP&quot; on page 249</td>
<td>Configure outgoing mail server</td>
<td></td>
</tr>
<tr>
<td>&quot;SNMP&quot; on page 250</td>
<td>SNMP for sending traps on alerts and serving appliance status information</td>
<td></td>
</tr>
<tr>
<td>&quot;Syslog&quot; on page 253</td>
<td>Syslog Relay for sending syslog messages on alerts and forwarding service syslog messages</td>
<td></td>
</tr>
<tr>
<td>&quot;System Identity&quot; on page 258</td>
<td>System name and location</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 53  Available Remote Access Services

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
<th>Ports Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;SSH&quot; on page 259</td>
<td>SSH for CLI access</td>
<td>22</td>
</tr>
<tr>
<td>&quot;REST&quot; on page 248</td>
<td>RESTful API</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 54  Available Security Services

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
<th>Ports Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kerberos</td>
<td>Kerberos V Authentication</td>
<td>88</td>
</tr>
<tr>
<td>Kerberos V Change &amp; Set Password (SET_CHANGE)</td>
<td>Kerberos V Change &amp; Set Password (RPCSEC_GSS)</td>
<td>464 749</td>
</tr>
</tbody>
</table>
Required Service Ports

To provide security on a network, you can deploy firewalls within your network architecture. Port numbers are used for creating firewall rules and uniquely identify a transaction over a network by specifying the host and the service.

The following list shows the minimum ports required for creating firewall rules that allow full functionality of the appliance:

**Inbound Ports**
- icmp/0-65535 (PING)
- tcp/1920 (EM)
- tcp/215 (BUI)
- tcp/22 (SSH)
- udp/161 (SNMP)

**Outbound Ports**
- tcp/80 (WEB)
- tcp/443 (SSL WEB)

**Note** - An outbound port of tcp/443 is used for sending Phone Home messages, uploading support bundles, and update notifications. For replication, use Generic Routing Encapsulation (GRE) tunnels when possible. This lets traffic run on the back end interfaces and avoid the firewall where traffic could be slowed. If GRE tunnels are not available on the NFS core, you must run replication over the front end interface. In this case, port 216 must also be open.

Services BUI Page Icons and Buttons

You use the BUI Services screens to view and modify the services and settings described in tables above. Double click a service line to view the definition screen for that service. The following tables describes the icons and buttons in the services screens:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Go to the service screen to configure properties and view logs. This button appears when you mouse-over a service.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>The service is enabled and working normally.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>The service is offline or disabled.</td>
</tr>
</tbody>
</table>
Working with Oracle ZFS Storage Appliance Services

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>The service has a problem and requires operator attention.</td>
</tr>
<tr>
<td>🌋</td>
<td>Enables or disables the service</td>
</tr>
<tr>
<td>⚡️</td>
<td>Restarts the service</td>
</tr>
<tr>
<td>🌋</td>
<td>Enable/disable not available for this service</td>
</tr>
<tr>
<td>⚡️</td>
<td>Restarts the currently unavailable service. You must enable the service first</td>
</tr>
</tbody>
</table>

## Configuring Services using the CLI

The CLI services section is under configuration services. Use the show command to list the current state of all services:

The following example is from the NTP service:

```
caji:> configuration services
caji:configuration services> show
Services:
    ad => disabled
    smb => disabled
    dns => online
    dynrouting => online
    ftp => disabled
    http => disabled
    identity => online
    idmap => online
    ipmp => online
    iscsi => online
    ldap => disabled
    ndmp => online
    nfs => online
    nis => disabled
    ntp => disabled
    replication => online
```

The first log event in the example shows that the system was booted at 21:05. The second entry at 21:07:37 records that the service began startup, which completed at 21:13:38. Due to the nature of NTP and system clock adjustment, this service can take minutes to complete startup, as shown by the log.
Viewing a Specific Service Screen using the BUI

1. To view or edit the properties for a specific service, mouse over the service the status icon that is to the left of the service name.

2. The status icon turns into an arrow icon, which you click to display the properties screen for the selected service.
Viewing a Specific Service Screen

- In any of the services screens, you can show a side panel of all services by clicking the small arrow icon to the left of the Services title (near the top left of each screen). Click this icon again to hide the list.

Enabling a Service using the BUI

- If a service is not online, click the power icon to bring the service online.

Disabling a Service using the BUI

- If a service is online and you want to disabled it, click the power icon to take the service offline.

Defining Properties using the BUI

1. To define properties for a service, double click a service.
2. Change the properties and then click APPLY.
3. To reset properties, click REVERT.

Viewing Service Logs using the BUI

1. Some services provide service logs with information to help you diagnose service issues. If a Logs button exists in the top right of a service screen, that service provides a log. Logs can provide the following information:
   - Times when a service changed state
   - Error messages from the service
2. Log content is specific to each individual service and is subject to change with future updates to the appliance software. The following are example messages that are commonly used in this version of the appliance:
Selecting a Service using the CLI

1. After you select a service, you can view its state, enable it, disable it, and set its properties.

2. Select a service by entering its name. For example, to select nis:

   caji:configuration services> nis
   caji:configuration services nis>

Viewing a Service's State using the CLI

- You can view a service's state using the show command:

   caji:configuration services nis> show
   Properties:
   <status> = online
   domain = fishworks
   broadcast = true
   ypservers =

Enabling a Service using the CLI

- Use the enable command to enable a service:
Disabling a Service using the CLI

- Use the `disable` command to disable a service:

```
caji:configuration services nis> disable
```

Setting Properties using the CLI

1. Use the `set` command to set the properties for the selected service.

```
caji:configuration services nis> set domain="mydomain"
     domain = mydomain (uncommitted)
```

2. After setting the properties, use the `commit` command to save and activate the new configuration:

```
caji:configuration services nis> commit
```

3. Property names are similar to their names in the BUI, but CLI names are usually shorter and sometimes abbreviated.

Viewing Service Help using the CLI

- Type `help` to see all commands for a service:

```
caji:configuration services nis> help
Subcommands that are valid in this context:

    help [topic]  => Get context-sensitive help. If [topic] is specified, it must be one of "builtins", "commands", "general", "help", "script" or "properties".
    show         => Show information pertinent to the current context
    commit       => Commit current state, including any changes
```
Working with NFS

Network File System (NFS) is an industry standard protocol to share files over a network. The Sun ZFS Storage Appliance supports NFS versions 2, 3, and 4. For more information on how the filesystem namespace is constructed, see “Working with Filesystem Namespace” on page 294. For information about NFS with local users, see “Configuring Users” on page 124.

To work with NFS, see the following sections:
- “NFS Properties” on page 183
- “Configuring Kerberos Realms for NFS” on page 185
- “NFS Logs” on page 186
- “NFS Analytics” on page 186
- “NFS Properties” on page 187
- “Sharing a Filesystem over NFS” on page 187

NFS Properties

- Minimum supported version - Use this drop-down list to control which versions of NFS the appliance supports.
- Maximum supported version - Use this drop-down list to control which versions of NFS the appliance supports.
- Maximum # of server threads - Define the maximum number of concurrent NFS requests (from 20 to 1000). This should at least cover the number of concurrent NFS clients that you anticipate.
- Grace period - Define the number of seconds that all clients have to recover locking state after an appliance reboot (from 15 to 600 seconds) from an unplanned outage. This
property affects only NFS v4 clients (NFS v3 is stateless so there is no state to reclaim). During this period, the NFS service only processes reclaims of the old locking state. No other requests for service are processed until the grace period is over. The default grace period is 90 seconds. Reducing the grace period lets NFS clients resume operation more quickly after a server reboot, but increases the probability that a client cannot recover all of its locking state. The Oracle ZFS Storage Appliance provides grace-less recovery of the locking state for NFSv4 clients during planned outages. Planned outages occur during events such as updates and appliance reboot using the CLI maintenance system reboot command or the BUI power icon.

For planned outages, the NFS service processes all requests for service without incurring the grace period delay.

- Custom NFSv4 identity domain - Use this property to define the domain for mapping NFSv4 users and group identities. If you do not set this property, the appliances uses DNS to obtain the identity domain, first by checking for a _nfsv4idmapdomain DNS resource record, and then by falling back to the DNS domain itself.

- Enable NFSv4 delegation - Select this property to allow clients to cache files locally and make modifications without contacting the server. This option is enabled by default and typically results in better performance; but in rare circumstances it can cause problems. You should only disable this setting after careful performance measurements of your particular workload and after validating that the setting has a measurable performance benefit. This option only affects NFSv4 mounts.

- Mount visibility - This property lets you limit the availability of information about share access lists and remote mounts from NFS clients. Full allows full access. Restricted restricts access such that a client can see only the shares which it is allowed to access. A client cannot see access lists for shares defined at the server or remote mounts from the server done by other clients. The property is set to Full by default.

- Enable Kerberos - Enables/disables Kerberos service.

  - Allow weak encryption types in Kerberos - Enables/disables support for DES (des-cbc-crc, des-cbc-md5) and Exportable ArcFour with HMAC/md5 (arcfour-hmac-exp). This property is disabled by default.

  - Kerberos realm - A realm is logical network, similar to a domain, that defines a group of systems that are under the same master KDC. Realm names can consist of any ASCII string. Usually, your realm name is the same as your DNS domain name, except that the realm name is in uppercase. Using this convention helps you differentiate problems with the Kerberos service from problems with the DNS namespace, while still using a name that is familiar.

  - Kerberos master KDC - In each realm, you must include a server that maintains the master copy of the principal database. The most significant difference between a master KDC and a slave KDC is that only the master KDC handles database administration requests. For instance, you must change a password or add a new principal on the master KDC.

  - Kerberos slave KDC - The slave contains duplicate copies of the principal database. Both the master KDC server and the slave KDC server create tickets that are used to establish authentication.
- Kerberos admin principal - This property identifies the administrator. By convention, a principal name is divided into three components: the primary, the instance, and the realm. You can specify a principal as joe, joe/admin, or joe/admin@ENG.EXAMPLE.COM. This property is used only to set up the system's Kerberos service principals and is not retained.
- Kerberos admin password - Defines a password for the administrator. This property is used only to set up the system's Kerberos service principals and is not retained.
- Oracle Intelligent Storage Protocol - The NFSv4 service includes support for the Oracle Intelligent Storage Protocol, which lets Oracle Database NFSv4 clients pass optimization information to the ZFS Storage Appliance NFSv4 server. For more information, see “Oracle Intelligent Storage Protocol” on page 445.

Changing services properties is documented in the “Services BUI Page Icons and Buttons” on page 177 and “Configuring Services using the CLI” on page 178.

Setting the NFS minimum and maximum versions to the same value causes the appliance to only communicate with clients using that version. This may be useful if you find an issue with one NFS version or the other (such as the performance characteristics of an NFS version with your workload), and you want to force clients to only use the version that works best.

### Configuring Kerberos Realms for NFS

Configuring a Kerberos realm creates certain service principals and adds the necessary keys to the system's local keytab. The NTP service must be configured before configuring Kerberized NFS. The following service principals are created and updated to support Kerberized NFS:

- host/node1.example.com@EXAMPLE.COM
- nfs/node1.example.com@EXAMPLE.COM

If you clustered your appliances, principals and keys are generated for each cluster node:

- host/node1.example.com@EXAMPLE.COM
- nfs/node1.example.com@EXAMPLE.COM
- host/node2.example.com@EXAMPLE.COM
- nfs/node2.example.com@EXAMPLE.COM

If these principals have already been created, configuring the realm resets the password for each of those principals. If you configured your appliance to join an Active Directory domain, you cannot configure it to be part of a Kerberos realm.

For information on setting up KDCs and Kerberized clients, see Oracle Solaris 11.1 Administration: Security Services (http://docs.oracle.com/cd/E26502_01/html/E29015/index.html). After setting NFS properties for Kerberos, change the Security mode on the Shares->Filesystem->Protocols screen to a mode using Kerberos.
The following ports are used by the appliance for Kerberos.

- Kerberos V authentication: 88
- Kerberos V change and set password SET_CHANGE: 464
- Kerberos V change and set password RPCSEC_GSS: 749

**Note** - Kerberized NFS clients must access the appliance using an IP address that resolves to an FQDN for those principals. For example, if an appliance is configured with multiple IP addresses, only the IP address that resolves to the appliance's FQDN can be used by its Kerberized NFS clients.

### NFS Logs

These logs are available for the NFS service:

**TABLE 56 Logs Available for NFS**

<table>
<thead>
<tr>
<th>Log</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network-nfs-server:default</td>
<td>Master NFS server log</td>
</tr>
<tr>
<td>appliance-kit-nfsconf:default</td>
<td>Log of appliance NFS configuration events</td>
</tr>
<tr>
<td>network-nfs-cbd:default</td>
<td>Log for the NFSv4 callback daemon</td>
</tr>
<tr>
<td>network-nfs-mapid:default</td>
<td>Log for the NFSv4 mapid daemon - which maps NFSv4 user and group credentials</td>
</tr>
<tr>
<td>network-nfs-status:default</td>
<td>Log for the NFS statd daemon - which assists crash and recovery functions for NFS locks</td>
</tr>
<tr>
<td>network-nfs-nlockmgr:default</td>
<td>Log for the NFS lockd daemon - which supports record locking operations for files</td>
</tr>
</tbody>
</table>

### NFS Analytics

You can monitor NFS activity in the Analytics section. This includes:

- NFS operations per second
- ... by type of operation (read/write/...)
- ... by share name
- ... by client hostname
- ... by accessed filename
- ... by access latency

**Note:** When the NFS server reboots or fails over the filename is *unknown* at the server until a new open from the client. The file appears as *unknown* in Analytics worksheets.
NFS Properties

The following table describes the mapping between CLI properties and the BUI property descriptions above.

<table>
<thead>
<tr>
<th>CLI Property</th>
<th>BUI Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>version_min</td>
<td>Minimum supported version</td>
</tr>
<tr>
<td>version_max</td>
<td>Maximum supported version</td>
</tr>
<tr>
<td>nfsd_servers</td>
<td>Maximum # of server threads</td>
</tr>
<tr>
<td>grace_period</td>
<td>Grace period</td>
</tr>
<tr>
<td>mapid_domain</td>
<td>Custom NFSv4 identity domain</td>
</tr>
<tr>
<td>enable_delegation</td>
<td>Enable NFSv4 delegation</td>
</tr>
<tr>
<td>mount_visibility</td>
<td>Client share information restriction level</td>
</tr>
<tr>
<td>krb5_allow_weak_crypto</td>
<td>Permits weak encryption types (arcfour-hmac-md5-exp, des-cbc-md5, and des-cbc-crc) in Kerberos</td>
</tr>
<tr>
<td>krb5_realmin</td>
<td>Kerberos Realm</td>
</tr>
<tr>
<td>krb5_kdc</td>
<td>Kerberos master KDC</td>
</tr>
<tr>
<td>krb5_kdc2</td>
<td>Kerberos slave KDC</td>
</tr>
<tr>
<td>krb5_admin</td>
<td>Kerberos admin principal</td>
</tr>
</tbody>
</table>

▼ Sharing a Filesystem over NFS

1. Go to the Configuration->Services screen.
2. Check that the NFS service is enabled and online. If not, enable the service.
3. Got to the Shares screen and edit an existing share or create a new share.
4. Click the Protocols tab of the share you are editing and check that NFS sharing is enabled. You can also configure the NFS share mode (read/read+write) in this screen.

Working with iSCSI

When you configure a LUN on the appliance you can export that volume over an Internet Small Computer System Interface (iSCSI) target. The iSCSI service allows iSCSI initiators to access targets using the iSCSI protocol.
The service supports discovery, management, and configuration using the iSNS protocol. The iSCSI service supports both unidirectional (target authenticates initiator) and bidirectional (target and initiator authenticate each other) authentication using CHAP. Additionally, the service supports CHAP authentication data management in a RADIUS database.

The system performs authentication first, and authorization second, in two independent steps.

**Note** - For examples of configuring iSCSI initiators and targets, see “Configuring Storage Area Network” on page 99.

Changing services properties is documented in “Services BUI Page Icons and Buttons” on page 177 and “Configuring Services using the CLI” on page 178. The CLI property names are shorter versions of those listed above.

<table>
<thead>
<tr>
<th>TABLE 58</th>
<th>iSCSI Service Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>Use iSNS</td>
<td>Whether iSNS discovery is enabled</td>
</tr>
<tr>
<td>iSNS Server</td>
<td>An iSNS server</td>
</tr>
<tr>
<td>Use RADIUS</td>
<td>Whether RADIUS is enabled</td>
</tr>
<tr>
<td>RADIUS Server</td>
<td>A RADIUS server</td>
</tr>
<tr>
<td>RADIUS Server Secret</td>
<td>The RADIUS server's secret</td>
</tr>
</tbody>
</table>

If the local initiator has a CHAP name and a CHAP secret, the system performs authentication. If the local initiator does not have the CHAP properties, the system does not perform any authentication and therefore all initiators are eligible for authorization.

The iSCSI service allows you to specify a global list of initiators that you can use within initiator groups.

For more information on iSCSI targets and initiators, see “Configuring Storage Area Network” on page 99.

If your initiator cannot connect to your target:

- Make sure the IQN of the initiator matches the IQN identified in the initiators list.
- Check that IP address of iSNS server is correct and that the iSNS server is configured.
- Check that the IP address of the target is correct on the initiator side.
- Check that initiator CHAP names and secrets match on both sides.
- Make sure that the target CHAP name and secret do not match those of any of the initiators.
- Check that the IP address and secret of the RADIUS server are correct, and that the RADIUS server is configured.
- Check that the initiator accessing the LUN is a member of that LUN's initiator group.
- Check that the targets exporting that LUN are online.
- Check that the LUN's operational status is online.
■ Check the logical unit number for each LUN.

If, during the failover / failbacks, the iSER Reduced Copy I/Os from the Red Hat client are not surviving, modify the node.session.timeo.replace_timeout parameter in the /etc/iscsi/iscsid.conf file to 300sec.

Working with SMB

The SMB service provides access to filesystems using the SMB protocol. The supported SMB versions are: SMB 1, SMB 2.0 and SMB 2.1. Filesystems must be configured to share using SMB from the Shares configuration. The following table shows the supported and unsupported features of SMB 2.1.

<table>
<thead>
<tr>
<th>SUPPORTED FEATURES</th>
<th>UNSUPPORTED FEATURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leases</td>
<td>Branch cache</td>
</tr>
<tr>
<td>Multi-protocol negotiate request</td>
<td>Resilient handles</td>
</tr>
<tr>
<td>Individual write-through operations</td>
<td>Multi-credit operations</td>
</tr>
</tbody>
</table>

Local accounts and user IDs are mapped to Windows user IDs. Note that the guest account is a special, readonly account and cannot be configured for read/write in the appliance.

To work with SMB, see the following sections:
■ “SMB Properties” on page 190
■ “SMB Share Properties” on page 191
■ “NFS/SMB Interoperability” on page 192
■ “SMB DFS Namespaces” on page 192
■ “SMB Microsoft Stand-alone DFS Namespace Management Tools Support Matrix” on page 193
■ “Adding DFS Namespaces to a Local SMB Group” on page 194
■ “SMB Autohome” on page 195
■ “Adding SMB Autohome Rules” on page 195
■ “SMB Local Groups” on page 196
■ “Adding a User to an SMB Local Group” on page 197
■ “SMB MMC Integration” on page 197
■ “SMB Share Management” on page 198
■ “SMB Users, Groups, and Connections” on page 199
■ “Listing SMB Services” on page 200
■ “Configuring SMB using the BUI” on page 202
■ “Configuring SMB Active Directory using the BUI” on page 203
“Configuring SMB Project and Share using the BUI” on page 204
“Configuring SMB Data Service using the BUI” on page 204

SMB Properties

- **LAN Manager compatibility level** - Authentication modes supported (LM, NTLM, LMv2, NTLMv2). For more information on the supported authentication modes within each compatibility level, consult the Oracle Solaris Information Library for `smb`. NTLMv2 is the recommended minimum security level to avoid publicly known security vulnerabilities.

- **Preferred domain controller** - The preferred domain controller to use when joining an Active Directory domain. If this controller is not available, Active Directory will rely on DNS SRV records and the Active Directory site to locate an appropriate domain controller. For more information, see “Working with Active Directory” on page 228.

- **Active Directory site** - The site to use when joining an Active Directory domain. A site is a logical collection of machines which are all connected with high bandwidth, low latency network links. When this property is configured and the preferred domain controller is not specified, joining an Active Directory domain will prefer domain controllers located in this site over external domain controllers.

- **Maximum # of server threads** - The maximum number of simultaneous server threads (workers). Default is 1024.

- **Enable Dynamic DNS** - Choose whether the appliance will use Dynamic DNS to update DNS records in the Active Directory domain. Default is off.

- **Enable Oplocks** - Choose whether the appliance will grant Opportunistic Locks to SMB clients. This will improve performance for most clients. Default is on. The SMB server grants an oplock to a client process so that the client can cache data while the lock is in place. When the server revokes the oplock, the client flushes its cached data to the server.

- **Restrict anonymous access to share list** - If this option is enabled, clients must authenticate to the SMB service before receiving a list of shares. If disabled, anonymous clients may access the list of shares.

- **System Comment** - Meaningful text string.

- **Idle Session Timeout** - Timeout setting for session inactivity.

- **Primary WINS server** - Primary WINS address configured in the TCP/IP setup.

- **Secondary WINS server** - Secondary WINS address configured in the TCP/IP setup.

- **Excluded IP addresses from WINS** - IP addresses excluded from registration with WINS.

- **SMB Signing Enabled** - Enables interoperability with SMB clients using the SMB signing feature. If a packet has been signed, the signature will be verified. If a packet has not been signed it will be accepted without signature verification (if SMB signing is not required - see below).
- **SMB Signing Required** - When SMB signing is required, all SMB packets must be signed or they will be rejected, and clients that do not support signing will be unable to connect to the server.

- **Ignore zero VC** - When an SMB client establishes a new connection, it may request that the appliance clean up all previous connections and file locks from this client by specifying a Virtual Circuit (VC) number of zero. This protocol artifact however, does not respect network address translation (NAT) for clients or multiple DNS entries assigned to the same host. In combination, zero VC requests between masked or redundant network locations may result in unrelated active connections being reset. By default, zero VC requests are honored to prevent stale file locking, however if SMB sessions are being disconnected in error, ignoring zero VC requests may resolve the issue.

Changing service properties is documented in “Services BUI Page Icons and Buttons” on page 177 and “Configuring Services using the CLI” on page 178. The CLI property names are shorter versions of those listed above.

## SMB Share Properties

Several share properties must be set in certain ways when exporting a share over SMB.

<table>
<thead>
<tr>
<th>TABLE 60</th>
<th>SMB Share Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>“Working with Shares”</td>
<td>SMB clients expect case-insensitive behavior, so this property must be “mixed” or “insensitive”.</td>
</tr>
<tr>
<td>“Working with Shares”</td>
<td>If non-UTF-8 filenames are allowed in a filesystem, SMB clients may function incorrectly.</td>
</tr>
<tr>
<td>Non-Blocking Mandatory Locking</td>
<td>This property must be enabled to allow byte range locking to function correctly.</td>
</tr>
<tr>
<td>“Understanding the Shares &gt; Shares &gt; Protocols BUI Page” on page 314</td>
<td>The name by which clients refer to the share. For information about how this name is inherited from a “Working with Shares”, see “Understanding the Shares &gt; Shares &gt; Protocols BUI Page” on page 314.</td>
</tr>
<tr>
<td>“Understanding the Shares &gt; Shares &gt; Protocols BUI Page” on page 314</td>
<td>An ACL which adds another layer of access control beyond the ACLs stored in the filesystem. For more information on this property, see “Understanding the Shares &gt; Shares &gt; Protocols BUI Page” on page 314.</td>
</tr>
</tbody>
</table>

Share properties can only be set when creating a share.
NFS/SMB Interoperability

The appliance supports NFS and SMB clients accessing the same shares concurrently. To correctly configure the appliance for NFS/SMB interoperability, you must configure the following components:

- Configure the Active Directory service. See “Working with Active Directory” on page 228.
- Establish an Identity Mapping strategy and configure the service. See “Working with Identity Mapping” on page 233.
- Configure SMB.
- Configure access control, ACL entries, and ACL inheritance on shares.

SMB and NFSv3 do not use the same access control model. For best results, configure the ACL on the root directory from a SMB client as the SMB access control model is a more verbose model. For information on inheritable trivial ACL entries, see “Working with the Shares > Shares > Access BUI Page” on page 321.

SMB DFS Namespaces

The Distributed File System (DFS) is a virtualization technology delivered over the SMB and MSRPC protocols. DFS allows administrators to group shared folders located on different servers by transparently connecting them to one or more DFS namespaces. A DFS namespace is a virtual view of shared folders in an organization. An administrator can select which shared folders to present in the namespace, design the hierarchy in which those folders appear and determine the names that the shared folders show in the namespace. When a user views the namespace, the folders appear to reside in a single, high-capacity file system. Users can navigate the folders in the namespace without needing to know the server names or shared folders hosting the data.

Only one share per system may be provisioned as a standalone DFS namespace. Domain-based DFS namespaces are not supported. Note that one DFS namespace may be provisioned per cluster, even if each cluster node has a separate storage pool. To provision a SMB share as a DFS namespace, use the DFS Management MMC Snap-in to create a standalone namespace.

When the appliance is not joined to an Active Directory domain, additional configuration is necessary to allow Workgroup users to modify DFS namespaces. To enable an SMB local user to create or delete a DFS namespace, that user must have a separate local account created on the server. In the example below, the steps let the SMB local user dʃsadmʃn manipulate DFS namespaces.
# SMB Microsoft Stand-alone DFS Namespace Management Tools Support Matrix

The following table lists operations (subcommands/options) of the Microsoft DFS tools on various Windows operating system versions. It identifies which of these are supported by the DFS service on the appliance for managing a standalone DFS namespace on the appliance.

- **y** - supported
- **n** - not supported
- **NA** - not applicable

<table>
<thead>
<tr>
<th>Microsoft Windows systems</th>
<th>XP</th>
<th>2003</th>
<th>R2</th>
<th>2003</th>
<th>Vista</th>
<th>2008</th>
<th>R2</th>
<th>2008</th>
<th>Win7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SP3</td>
<td>SP2</td>
<td>SP2</td>
<td>SP2</td>
<td>SP2</td>
<td>SP1</td>
<td>SP1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dfscmd CLI:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/map [comment] [restore]</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/unmap</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/add [restore]</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/remove</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/view [partial</td>
<td>full]</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dfsutil CLI (old format):</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/addstdroot [comment]</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/remstdroot</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/root:&lt;DfsName&gt; /view</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/addlink [comment]</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/removelnk</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/state /display</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/state /enable</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/state /disable</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/ttl /display</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/ttl /set</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/server:&lt;MachineName&gt; /view</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dfsutil CLI (new format):</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>root addstd [comment]</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>root remove</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>root (view namespace)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link add [comment]</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link remove</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link (view)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>target add</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>target remove</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>target (view)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>property comment (view)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Adding DFS Namespaces to a Local SMB Group

<table>
<thead>
<tr>
<th>Property</th>
<th>XP</th>
<th>2003</th>
<th>R2</th>
<th>SP3</th>
<th>SP2</th>
<th>SP1</th>
<th>SP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Comment Set</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>Property TTL View</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>Property TTL Set</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>Property State View</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>Property State Offline</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>Property State Online</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
</tbody>
</table>

**DFS GUI:**

<table>
<thead>
<tr>
<th></th>
<th>2003</th>
<th>Vista</th>
<th>2008</th>
<th>Win7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Standalone Root</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Remove Standalone Root</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Change Root Comment</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Change Root Timeout</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Add Link</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Remove Link</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Change Link Comment</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Change Link Timeout</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Add Link's Target</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Remove Link's Target</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Enable Link's Referral</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Disable Link's Referral</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Hide Root</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Show Root</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Display Links</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>Display Targets</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
</tbody>
</table>

Note that:

- Solaris does not verify the DFS link target.
- CLI commands for modifying and viewing comment and timeout (TTL) are applicable to both root and link.
- CLI commands for viewing state are applicable to root, root's target, link, and link's target.
- CLI commands for modifying state are only applicable for link and link's target.

#### Adding DFS Namespaces to a Local SMB Group

1. **Create a local user account on the server for user `dfsadmin`**. Be sure to use the same password as when the local user was first created on the Windows machine.

2. **Add `dfsadmin` to the local SMB group Administrators**.

3. **Login as `dfsadmin` on the Windows machine from which the DFS namespace will be modified**.
**SMB Autohome**

For Windows file sharing, Autohome provides access to filesystems using the SMB protocol. Autohome defines and maintains home directory shares for users that access the system through SMB. Autohome rules map SMB clients to home directories.

FIGURE 33  Setting Autohome Rules

- **Use Name Service Switch** - Toggles Name Service Switch (NSS) on or off. You cannot create an NSS rule and an rule for all users at the same time.
- **AD Container** - Sets the Active Directory container, for example: dc=com,dc=fishworks, ou=Engineering,CN=myhome.
- **User** - Sets the Autohome rule for all All users or for the user you specify. When you specify a user, the wildcards "&" and "?" refer respectively to each user's login and its corresponding first character.
- **Directory** - Sets the directory for the rule, for example: /export/wdp.

▼ **Adding SMB Autohome Rules**

1. Use the `create` command to add autohome rules, and the `list` command to list existing rules. This example adds a rule for the user "Bill" then lists the rules:

    ```
    twofish:> configuration services smb
    ```
Adding SMB Autohome Rules

twofish:configuration services smb> create
twofish:configuration services rule (uncommitted)> set use_nss=false
twofish:configuration services rule (uncommitted)> set user=Bill
twofish:configuration services rule (uncommitted)> set directory=/export/wdp
twofish:configuration services rule (uncommitted)> set container="dc=com,dc=fishworks, ou=Engineering,CN=myhome"
twofish:configuration services rule (uncommitted)> commit
twofish:configuration services smb> list

RULE       NSS      USER         DIRECTORY            CONTAINER
rule-000   false    Bill         /export/wdp          dc=com,dc=fishworks, ou=Engineering,CN=myhome

2. **Autohome rules may be created using wildcard characters. The & character matches the users' username, and the ? character matches the first letter of the users' username. The following uses wildcards to match all users:**

twofish:configuration services smb> create
twofish:configuration services rule (uncommitted)> set use_nss=false
twofish:configuration services rule (uncommitted)> set user=*
twofish:configuration services rule (uncommitted)> set directory=/export/??&
twofish:configuration services rule (uncommitted)> set container="dc=com,dc=fishworks, ou=Engineering,CN=myhome"
twofish:configuration services rule (uncommitted)> commit
twofish:configuration services smb> list

RULE       NSS      USER         DIRECTORY            CONTAINER
rule-000   false    Bill         /export/wdp          dc=com,dc=fishworks, ou=Engineering,CN=myhome

3. **The name service switch may also be used to create autohome rules:**

twofish:configuration services smb> create
twofish:configuration services rule (uncommitted)> set use_nss=true
twofish:configuration services rule (uncommitted)> set container="dc=com,dc=fishworks, ou=Engineering,CN=myhome"
twofish:configuration services rule (uncommitted)> commit
twofish:configuration services smb> list

RULE       NSS      USER         DIRECTORY            CONTAINER
rule-000   true                                       dc=com,dc=fishworks, ou=Engineering,CN=myhome

SMB Local Groups

Local groups are groups of domain users which confer additional privileges to those users.

**TABLE 61** SMB Local Groups

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrators</td>
<td>Administrators can bypass file permissions to change the ownership on files.</td>
</tr>
</tbody>
</table>
Adding a User to an SMB Local Group

To add a user to an SMB local group, use the following CLI commands.

```
twofish:configuration services smb> groups
twofish:configuration services smb groups> create
twofish:configuration services smb member (uncommitted)> set user=Bill
twofish:configuration services smb member (uncommitted)> set group="Backup Operators"
twofish:configuration services smb member (uncommitted)> commit
twofish:configuration services smb groups> list
```

MEMBER USER GROUP
member-000 WINDOMAIN\Bill Backup Operators

SMB MMC Integration

The Microsoft Management Console (MMC) is an extensible framework of registered components, known as snap-ins, that provide comprehensive management features for both the local system and remote systems on the network. Computer Management is a collection of Microsoft Management Console tools, that may be used to configure, monitor and manage local and remote services and resources.

In order to use the MMC functionality on the Oracle ZFS Storage appliances in workgroup mode, be sure to add the Windows administrator who will use the management console to the Administrators local group on the appliance. Otherwise you may receive an Access is denied or similar error on the administration client when attempting to connect to the appliance using the MMC.

The Oracle ZFS Storage appliances support the following Computer Management facilities:

Display of the Application log, Security log, and System log are supported using the Event Viewer MMC snap-in. These logs show the contents of the alert, audit, and system logs of the Oracle ZFS Storage system. Following is a screen capture that illustrates the Application log and the properties dialog for an error event.
Adding a User to an SMB Local Group

SMB Share Management

Support for share management includes the following:

- Listing shares
- Setting ACLs on shares
- Changing share permissions
- Setting the description of a share

Features not currently supported via MMC include the following:

- Adding or Deleting a share
Adding a User to an SMB Local Group

- Setting client side caching property
- Setting maximum allowed or number of users property

**FIGURE 35** SMB Share Permission Properties

---

**SMB Users, Groups, and Connections**

The following features are supported:

- Viewing local SMB users and groups
- Listing user connections, including listing the number of open files per connection
- Closing user connections
Adding a User to an SMB Local Group

- Listing open files, including listing the number of locks on the file and file open mode
- Closing open files

**FIGURE 36** Open Files per Connection

![Open Files per Connection](image1)

**FIGURE 37** Open Sessions

![Open Sessions](image2)

**Listing SMB Services**

Support includes listing of appliance services. Services cannot be enabled or disabled using the Computer Management MMC application. Following is a screen capture that illustrates General properties for the vsan Service.
To ensure that only the appropriate users have access to administrative operations there are some access restrictions on the operations performed remotely using MMC.
TABLE 62  Users and Allowed Operations

<table>
<thead>
<tr>
<th>USERS</th>
<th>ALLOWED OPERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular users</td>
<td>List shares.</td>
</tr>
<tr>
<td>Members of the Administrators or Power Users groups</td>
<td>Manage shares, list user connections.</td>
</tr>
<tr>
<td>Members of the Administrators group</td>
<td>List open files and close files, disconnect user connections, view services and event log.</td>
</tr>
</tbody>
</table>

Configuring SMB using the BUI

Initial configuration of the appliance may be completed using the BUI or the CLI and should take less than 20 minutes. Initial Setup may also be performed again later using the Maintenance > System contexts of the BUI or CLI. Initial configuration will take you through the following BUI steps, in general.

1. Configure Network Devices, Datalinks, and Interfaces.

2. Create interfaces using the Datalink add or Interface icons or by using drag-and-drop of devices to the datalink or interface lists.

3. Set the desired properties and click the Apply button to add them to the list.

4. Set each interface to active or standby as appropriate.

5. Click the Apply button at the top of the page to commit your changes.

6. Configure DNS.

7. Provide the base domain name.

8. Provide the IP address of at least one server that is able to resolve hostname and server records in the Active Directory portion of the domain namespace.

9. Configure NTP authentication keys to ensure clock synchronization.

10. Click the icon to add a new key.

11. Specify the number, type, and private value for the new key and apply the changes. The key appears as an option next to each specified NTP server.

12. Associate the key with the appropriate NTP server and apply the changes. To ensure clock synchronization, configure the appliance and the SMB clients to use the same NTP server.
13. Specify Active Directory as the directory service for users and groups.

14. Set the directory domain.

15. Click the Apply button to commit your changes.

16. Configure a storage pool.

17. Click the icon to add a new pool.

18. Set the pool name.

19. On the "Allocate and verify storage" screen, configure the disk shelf allocation for the storage pool. If no disk shelves are detected, check your disk shelf cabling and power.

20. Click the Commit button to advance to the next screen.

21. On the "Configure Added Storage" screen, select the desired data profile. Each is rated in terms of availability, performance and capacity. Use these ratings to determine the best configuration for your business needs.

22. Click the Commit button to activate the configuration.


24. If the appliance is not directly connected to the internet, configure an HTTP proxy through which the remote support service may communicate with Oracle.

25. Enter your Online Account user name and password. A privacy statement will be displayed for your review.

26. Choose which of your inventory teams to register with. The default team for each account is the same as the account user name, prefixed with a '$'.

27. Commit your initial configuration changes.

▼ Configuring SMB Active Directory using the BUI

1. Create an account for the appliance in the Active Directory domain. Refer to Active Directory documentation for detailed instructions.

2. On the Configuration > Services > Active Directory screen, click the Join Domain button.
3. Specify the Active Directory domain, administrative user, administrative password and click the Apply button to commit the changes.

### Configuring SMB Project and Share using the BUI

1. Create a Project.
2. On the Shares screen, click the icon to expand the Projects panel.
3. Click the Add... link to add a new project.
4. Specify the Project name and apply the change.
5. Select the new project from the Projects panel.
6. Click the icon to add a filesystem.
7. Click the icon for the filesystem.
8. Click the General link and deselect the Inherit from project checkbox.
9. Choose a mountpoint under /export, even though SMB shares are accessed by resource name.
10. On the Protocols screen for the project, set the resource name to on.
11. Enable sharesmb and share-level ACL for the Project.
12. Click the Apply button to activate the configuration.

### Configuring SMB Data Service using the BUI

1. On the Configuration > Services > SMB screen, click the icon to enable the service.
2. Set SMB properties according to the recommendations in the properties section of this page and click the Apply button to activate the configuration.
3. Click the Autohome link on the Configuration > Services > SMB screen to set autohome rules to map SMB clients to home directories according to the
descriptions in the Autohome rules section above and click the Apply button to activate the configuration.

4. Click the Local Groups link on the Configuration > Services > SMB screen and use the icon to add administrators or backup operator users to local groups according to the descriptions in the Local Groups section above and click the Apply button to activate the configuration.

Working with FTP

The FTP (File Transfer Protocol) service allows filesystem access from FTP clients. Anonymous logins are not allowed, users must authenticate with whichever name service is configured in Services.

FTP Properties

<table>
<thead>
<tr>
<th>TABLE 63 FTP General Properties</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port (for incoming connections)</td>
<td>The port FTP listens on. Default is 21</td>
<td></td>
</tr>
<tr>
<td>Maximum # of connections (&quot;0&quot; for unlimited)</td>
<td>This is the maximum number of concurrent FTP connections. Set this to cover the anticipated number of concurrent users. By default this is 30, since each connection creates a system process and allowing too many (thousands) could constitute a DoS attack</td>
<td></td>
</tr>
<tr>
<td>Turn on delay engine to prevent timing attacks</td>
<td>This inserts small delays during authentication to fool attempts at user name guessing via timing measurements. Turning this on will improve security</td>
<td></td>
</tr>
<tr>
<td>Default login root</td>
<td>The FTP login location. The default is &quot;/&quot; and points to the top of the shares hierarchy. All users will be logged into this location after successfully authenticating with the FTP service</td>
<td></td>
</tr>
<tr>
<td>Logging level</td>
<td>The verbosity of the proftpd log.</td>
<td></td>
</tr>
<tr>
<td>Permissions to mask from newly created files and dirs</td>
<td>File permissions to remove when files are created. Group and world write are masked by default, to prevent recent uploads from being writeable by everyone</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE 64 FTP Security Properties</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable SSL/TLS</td>
<td>Allow SSL/TLS encrypted FTP connections. This will ensure that the FTP transaction is encrypted. Default is disabled.</td>
<td></td>
</tr>
</tbody>
</table>

Working with Services
Allowing FTP Access to a Share using the BUI

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port for incoming SSL/TLS connections</td>
<td>The port that the SSL/TLS encrypted FTP service listens on. Default is 21.</td>
</tr>
<tr>
<td>Permit root login</td>
<td>Allow FTP logins for the root user. This is off by default, since FTP authentication is plain text which poses a security risk from network sniffing attacks</td>
</tr>
<tr>
<td>Maximum # of allowable login attempts</td>
<td>The number of failed login attempts before an FTP connection is disconnected, and the user must reconnect to try again. By default this is 3</td>
</tr>
<tr>
<td>Permit foreign data connection addresses</td>
<td>Permits foreign FTP connections to enable direct transfer of files between FTP servers. This property is off by default.</td>
</tr>
</tbody>
</table>

FTP Logs

<table>
<thead>
<tr>
<th>Log</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>proftpd</td>
<td>Logs FTP events, including successful logins and unsuccessful login attempts</td>
</tr>
<tr>
<td>proftpd_xfer</td>
<td>File transfer log</td>
</tr>
<tr>
<td>proftpd_tls</td>
<td>Logs FTP events related to SSL/TLS encryption</td>
</tr>
</tbody>
</table>

Allowing FTP Access to a Share using the BUI

1. Go to Configuration->Services
2. Ensure that the FTP service is enabled and online. If not, enable the service.
3. Select or add a share in the Shares screen.
4. Go to the "Protocols" section, and check that FTP access is enabled. This is also where the mode of access (read/read+write) can be set.

Working with HTTP

The HTTP service provides access to filesystems using the HTTP and HTTPS protocols and the HTTP extension WebDAV (Web based Distributed Authoring and Versioning). This allows clients to access shared filesystems through a web browser, or as a local filesystem if their client software supports it. The URL to access these HTTP and HTTPS shares have the following formats respectively:

- http://hostname/shares/mountpoint/share_name
- https://hostname/shares/mountpoint/share_name
- The HTTPS server uses a self-signed security certificate.

To work with HTTP, see the following sections:
- “HTTP Properties” on page 207
- “HTTP Authentication and Access Control” on page 207
- “HTTP Logs” on page 208
- “Allowing HTTP Access to a Share using the BUI” on page 208

**HTTP Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Require client login</td>
<td>Clients must authenticate before share access is allowed, and files they create will have their ownership. If this is not set, files created will be owned by the HTTP service with user &quot;nobody&quot;. See the section on authentication below.</td>
</tr>
<tr>
<td>Protocols</td>
<td>Select which access methods to support HTTP, HTTPS, or both.</td>
</tr>
<tr>
<td>HTTP Port (for incoming connections)</td>
<td>HTTP port, default is 80</td>
</tr>
<tr>
<td>HTTPS Port (for incoming secure connections)</td>
<td>HTTP port, default is 443</td>
</tr>
</tbody>
</table>

**HTTP Authentication and Access Control**

If the "Require client login" option is enabled, then the appliance will deny access to clients that do not supply valid authentication credentials for a local user, a NIS user, or an LDAP user. Active Directory authentication is not supported.

Only basic HTTP authentication is supported. Note that unless HTTPS is being used, this transmits the username and password unencrypted, which may not be appropriate for all environments.

Normally, authenticated users have the same permissions with HTTP that they would have with NFS or FTP. Files and directories created by an authenticated user will be owned by that user, as viewed by other protocols. Privileged users (those having a uid less than 100) will be treated as "nobody" for the purposes of access control. Files created by privileged users will be owned by "nobody".

If the "Require client login" option is disabled, then the appliance will not try to authenticate clients (even if they do supply credentials). Newly created files are owned by "nobody", and all users are treated as "nobody" for the purposes of access control.
Regardless of authentication, no permissions are masked from created files and directories. Created files have Unix permissions 666 (readable and writable by everyone), and created directories have Unix permissions 777 (readable, writable, and executable by everyone).

**HTTP Logs**

<table>
<thead>
<tr>
<th>Log</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network-http:apache22</td>
<td>HTTP service log</td>
</tr>
</tbody>
</table>

▼ **Allowing HTTP Access to a Share using the BUI**

1. Go to Configuration->Services
2. Check that the HTTP service is enabled and online. If not, enable the service.
3. Select or add a share in the Shares screen.
4. Go to the "Protocols" section, and check that HTTP access is enabled. This is also where the mode of access (read/read+write) can be set.

**Working with NDMP**

The NDMP (Network Data Management Protocol) service enables the system to participate in NDMP-based backup and restore operations controlled by a remote NDMP client called a Data Management Application (DMA). Using NDMP, appliance user data (i.e., data stored in administrator-created shares on the appliance) can be backed up and restored to both locally attached tape devices and remote systems. Locally-attached tape devices can also be exposed to the DMA for backing up and restoring remote systems.

NDMP cannot be used to backup and restore system configuration data. Instead, use the Configuration Backup and Restore feature (see “Configuration Backup” in “Oracle ZFS Storage Appliance Customer Service Manual”).

To work with NDMP, see the following sections:

- “NDMP Local vs. Remote Configurations” on page 209
- “NDMP Backup Formats and Types ” on page 209
- “NDMP Back up with “dump” and “tar”” on page 210
NDMP Local vs. Remote Configurations

The appliance supports backup and restore using both a local configuration, in which tape drives are physically attached to the appliance, and a remote configuration, in which data is streamed to another system on the same network. In both cases, the backup must be managed by a supported DMA.

In local configurations, supported tape devices, including both drives and changers (robots), are physically connected to the system using a supported SCSI or Fibre Channel (FC) card configured in Initiator mode. These devices can be viewed on the NDMP Status screen. The NDMP service presents these devices to a DMA when the DMA scans for devices. Once configured in the DMA, these devices are available for backup and restore of the appliance or other systems on the same network. After adding tape drives or changers to the system or removing such devices from the system, a reboot may be required before the changes will be recognized by the NDMP service. After that, the DMA may need to be reconfigured because tape device names may have changed.

In remote configurations, the tape devices are not physically connected to the system being backed up and restored (the data server) but rather to the system running the DMA or a separate system (the tape server). These are commonly called "3-way configurations" because the DMA controls two other systems. In these configurations the data stream is transmitted between the data server and the tape server over an IP network.

NDMP Backup Formats and Types

The NDMP protocol does not specify a backup data format. The appliance supports three backup types corresponding to different implementations and on-tape formats. DMAs can select a backup type using the following values for the NDMP environment variable “TYPE”:

<table>
<thead>
<tr>
<th>Backup type</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>dump</td>
<td>File-based for filesystems only. Supports file history and direct access recovery (DAR).</td>
</tr>
<tr>
<td>tar</td>
<td>File-based for filesystems only. Supports file history and direct access recovery (DAR).</td>
</tr>
</tbody>
</table>
| zfs         | Share-based for both filesystems and volumes. Does not support file history or direct access recovery (DAR), but
Allowing HTTP Access to a Share using the BUI

<table>
<thead>
<tr>
<th>Backup type</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>may be faster for some datasets. Only supported with NDMPv4.</td>
</tr>
</tbody>
</table>

There is no standard NDMP data stream format, so backup streams generated on the appliance can only be restored on 7000-series appliances running compatible software. Future versions of appliance software can generally restore streams backed up from older versions of the software, but the reverse is not necessarily true. For example, the "zfs" backup type is new in 2010.Q3 and systems running 2010.Q1 or earlier cannot restore backup streams created using type "zfs" under 2010.Q3.

**NDMP Back up with "dump" and "tar"**

When backing up with "dump" and "tar" backup types, administrators specify the data to backup by a filesystem path, called the backup path. For example, if the administrator configures a backup of /export/home, then the share mounted at that path will be backed up. Similarly, if a backup stream is restored to /export/code, then that's the path where files will be restored, even if they were backed up from another path.

Only paths which are mountpoints of existing shares or contained within existing shares may be specified for backup. If the backup path matches a share's mountpoint, only that share is backed up. Otherwise the path must be contained within a share, in which case only the portion of that share under that path is backed up. In both cases, other shares mounted inside the specified share under the backup path will not be backed up; these shares must be specified separately for backup.

Snapshots - If the backup path specifies a live filesystem (e.g., /export/code) or a path contained within a live filesystem (e.g., /export/code/src), the appliance immediately takes a new snapshot and backs up the given path from that snapshot. When the backup completes, the snapshot is destroyed. If the backup path specifies a snapshot (e.g., /export/code/zfs/snapshot/mysnap), no new snapshot is created and the system backs up from the specified snapshot.

Share metadata - To simplify backup and restore of complex share configurations, "dump" and "tar" backups include share metadata for projects and shares associated with the backup path. This metadata describes the share configuration on the appliance, including protocol sharing properties, quota properties, and other properties configured on the Shares screen. This is not to be confused with filesystem metadata like directory structure and file permissions, which is also backed up and restored with NDMP.

For example, if you back up /export/proj, the share metadata for all shares whose mountpoints start with /export/proj will be backed up, as well as the share metadata for their parent projects. Similarly, if you back up /export/someshare/somedir, and a share is mounted at /export/someshare, that share and its project's share metadata will be backed up.

When restoring, if the destination of the restore path is not contained inside an existing share, projects and shares in the backup stream will be recreated as needed with their original
properties as stored in the backup. For example, if you back up /export/foo, which contains project proj1 and shares share1 and share2, and then destroy the project and restore from the backup, then these two shares and the project will be recreated with their backed-up properties as part of the restore operation.

During a restore, if a project exists that would have been automatically recreated, the existing project is used and no new project is automatically created. If a share exists that would have been automatically recreated, and if its mountpoint matches what the appliance expects based on the original backup path and the destination of the restore, then the existing share is used and no new share is automatically created. Otherwise, a new share is automatically created from the metadata in the backup. If a share with the same name already exists (but has a different mountpoint), then the newly created share will be given a unique name starting with "ndmp-" and with the correct mountpoint.

It is recommended that you either restore a stream whose datasets no longer exist on the appliance, allowing the appliance to recreate datasets as specified in the backup stream, or precreate a destination share for restores. Either of these practices avoids surprising results related to the automatic share creation described above.

**NDMP Back up with "zfs"**

When backing up with type "zfs", administrators specify the data to backup by its canonical name on the appliance. This can be found underneath the name of the share in the BUI:

![NDMP Share Name](image)

or in the CLI as the value of the canonical_name property. Canonical names do not begin with a leading '/', but when configuring the backup path the canonical name must be prefixed with '/'.

Both projects and shares can be specified for backup using type "zfs". If the canonical name is specified as-is, then a new snapshot is created and used for the backup. A specific snapshot can
be specified for backup using the '@snapshot' suffix, in which case no new snapshot is created and the specified snapshot is backed up. For example:

**TABLE 69**

<table>
<thead>
<tr>
<th>Canonical name</th>
<th>Shares backed up</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool-0/local/default</td>
<td>New snapshot of the local project called &quot;default&quot; and all of its shares.</td>
</tr>
<tr>
<td>pool-0/local/default@yesterday</td>
<td>Named snapshot &quot;yesterday&quot; of local project &quot;default&quot;, and all of its shares having snapshot &quot;yesterday&quot;.</td>
</tr>
<tr>
<td>pool-0/local/default/code</td>
<td>New snapshot of share &quot;code&quot; in local project &quot;default&quot;. &quot;code&quot; could be a filesystem or volume.</td>
</tr>
<tr>
<td>pool-0/local/default/code@yesterday</td>
<td>Named snapshot &quot;yesterday&quot; of share &quot;code&quot; in local project &quot;default&quot;. &quot;code&quot; could be a filesystem or volume.</td>
</tr>
</tbody>
</table>

Because level-based incremental backups using the "zfs" backup type require a base snapshot from the previous incremental, the default behavior for level backups for which a new snapshot is created is to keep the new snapshot so that it can be used for subsequent incremental backups. If the DMA indicates that the backup will not be used for subsequent incremental backups by setting UPDATE=n, the newly created snapshot is destroyed after the backup. Existing user snapshots are never destroyed after a backup. See "Incremental backups" below for details.

Share metadata - Share metadata (i.e., share configuration) is always included in "zfs" backups. When restoring a full backup with type "zfs", the destination project or share must not already exist. It will be recreated from the metadata in the backup stream. When restoring an incremental backup with type "zfs", the destination project or share must already exist. Its properties will be updated from the metadata in the backup stream. See "Incremental backups" below for details.

**NDMP Incremental backups**

The appliance supports level-based incremental backups for all of the above backup types. To specify a level backup, DMAs typically specify the following three environment variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td>Integer from 0 to 9 identifying the backup level.</td>
</tr>
<tr>
<td>DMP_NAME</td>
<td>Specifies a particular incremental backup set. Multiple sets of level incremental backups can be used concurrently by specifying different values for DMP_NAME.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Indicates whether this backup can be used as the base for subsequent incremental backups</td>
</tr>
</tbody>
</table>
By definition, a level-N backup includes all files changed since the previous backup of the
same backup set (specified by "DMP_NAME") of the same share using LEVEL less than
N. Level-0 backups always include all files. If UPDATE has value "y" (the default), then the
current backup is recorded so that future backups of level greater than N will use this backup as
a base. These variables are typically managed by the DMA and need not be configured directly
by administrators.

Below is a sample incremental backup schedule:

<table>
<thead>
<tr>
<th>TABLE 70 Sample Incremental Backup Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day</td>
</tr>
<tr>
<td>First of month</td>
</tr>
</tbody>
</table>
| Every 7th, 14th, 21st of month              | Level-1 backup. Backup contains all files changed since
|                                             | the last full (monthly) backup |
| Every day                                   | Level-2 backup. Backup contains all files changed since
|                                             | the last level-1 backup |

To recover the filesystem's state as it was on the 24th of the month, an administrator typically
restores the Level-0 backup from the 1st of the month to a new share, then restores the Level-1
backup from the 21st of the month, and then restores the Level-2 backup from the 24th of the
month.

To implement level-based incremental backups the appliance must keep track of the level
backup history for each share. For "tar" and "dump" backups, the level backup history is
maintained in the share metadata. Incremental backups traverse the filesystem and include
files modified since the time of the previous level backup. At restore time, the system simply
restores all the files in the backup stream. In the above example, it would therefore be possible
to restore the Level-2 backup from the 24th onto any filesystem and the files contained in that
backup stream will be restored even though the target filesystem may not match the filesystem
where the files were backed up. However, best practice suggests using a procedure like the
above which starts from an empty tree restores the previous level backups in order to recover
the original filesystem state.

To implement efficient level-based incremental backups for type "zfs", the system uses a
different approach. Backups that are part of an incremental set do not destroy the snapshot
used for the backup but rather leave it on the system. Subsequent incremental backups use this
snapshot as a base to quickly identify the changed filesystem blocks and generate the backup
stream. As a consequence, the snapshots left by the NDMP service after a backup must not be
destroyed if you want to create subsequent incremental backups.

Another important consequence of this behavior is that in order to restore an incremental
stream, the filesystem state must exactly match its state at the base snapshot of the incremental
stream. In other words, in order to restore a level-2 backup, the filesystem must look exactly
as it did when the previous level-1 backup completed. Note that the above commonly-used
procedure guarantees this because when restoring the Level-2 backup stream from the 24th,
the system is exactly as it was when the Level-1 backup from the 21st completed because that backup has just been restored.

The NDMP service will report an error if you attempt to restore an incremental "zfs" backup stream to a filesystem whose most recent snapshot doesn't match the base snapshot for the incremental stream, or if the filesystem has been changed since that snapshot. You can configure the NDMP service to rollback to the base snapshot just before the restore begins by specifying the NDMP environment variable "ZFS_FORCE" with value "y" or by configuring the "Rollback datasets" property of the NDMP service (see Properties below).

**NDMP Properties**

The NDMP service configuration consists of the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>The version of NDMP that your DMA supports.</td>
</tr>
<tr>
<td>TCP port (v4 only)</td>
<td>The NDMP default connection port is 10000. NDMPv3 always uses this port. NDMPv4 allows a different port if needed.</td>
</tr>
<tr>
<td>Default restore pool(s)</td>
<td>When you perform a full restore using &quot;tar&quot; or &quot;dump&quot;, the system re-creates datasets if there is no share mounted at the target. Because the NDMP protocol specifies only the mount point, the system chooses a pool in which to recreate projects and shares. On a system with multiple pools, this property lets you specify one or more pools. Multiple pools only need to be specified in a cluster with active pools on each head. You must ensure that this list is kept in sync with any storage configuration changes. If none of the pools exist or are online, the system will select a default pool at random.</td>
</tr>
<tr>
<td>Ignore metadata-only changes</td>
<td>Directs the system to backup only files in which content has changed, ignoring files for which only metadata, such as permissions or ownership, has changed. This option only applies to incremental &quot;tar&quot; and &quot;dump&quot; backups and is disabled by default.</td>
</tr>
<tr>
<td>Allow token-based backup</td>
<td>Enables or disables token-based method for ZFS backup. This property is off by default.</td>
</tr>
<tr>
<td>ZFS rollback before restore (v4 only)</td>
<td>Only applies to backups with type &quot;zfs&quot;. Determines whether when restoring an incremental backup the system rolls back the target project and share to the snapshot used as the base for the incremental restore. If the project and shares are rolled back, then any changes made since that snapshot will be lost. This setting is normally controlled by the DMA via the &quot;ZFS_FORCE&quot; environment variable (see &quot;Incremental Backups&quot; above) but this property can be used to override the DMA setting to always rollback these data sets or never roll them back. Not rolling them back will cause the restore...</td>
</tr>
</tbody>
</table>
Working with Remote Replication

When this service is enabled, the appliance will receive replication updates from other appliances as well as send replication updates for local projects and shares according to their configured actions. When the service is disabled, incoming replication updates will fail and no local projects and shares will be replicated.

This service doesn't have any properties, but it does allow administrators to view the appliances which have replicated data to this appliance (under Sources) and configure the appliances to which this appliance can replicate (under Targets).
Working with Shadow Migration

The shadow migration service allows for automatic migration of data from external or internal sources. This functionality is described in detail in “Working with Shadow Migration” on page 374. The service itself only controls automatic background migration. Regardless of whether the service is enabled or not, data will be migrated synchronously for in-band requests.

The service should only be disabled for testing purposes, or if the load on the system due to shadow migration is too great. When disabled, no filesystems will ever finish migrating. The primary purpose of the service is to allow tuning of the number of threads dedicated to background migration.

TABLE 73  Shadow Migration Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Threads</td>
<td>Number of threads to devote to background migration of data. These threads are global to the entire machine, and increasing the number can increase concurrency and the overall speed of migration at the expense of increased resource consumption (network, I/O, and CPU).</td>
</tr>
</tbody>
</table>

Working with SFTP

The SFTP (SSH File Transfer Protocol) service allows filesystem access from SFTP clients. Anonymous logins are not allowed, users must authenticate with whichever name service is configured in Services.

To work with SFTP, see the following sections:

- “SFTP Properties” on page 216
- “SFTP Port” on page 217
- “SFTP Logs” on page 217
- “Allowing SFTP Access to a Share using the BUI” on page 217
- “Configuring SFTP for Remote Access using the CLI” on page 217

SFTP Properties

- **Port** (for incoming connections) - The port SFTP listens on. Default is 218
- **Permit root login** - Allows SFTP logins for the root user. This is off by default.
- **Logging level** - The verbosity of SFTP log messages
- **SFTP Keys** - RSA/DSA public keys for SFTP authentication. Text comments can be associated with the keys to help administrators track why they were added. As of the
2011.1 software release, key management for SFTP has changed to increase security. When creating an SFTP key, it is required to include the "user" property with a valid user assignment. SFTP keys are grouped by user and are authenticated via SFTP with the user's name. It is recommended to recreate any existing SFTP keys that do not include the user property, even though they will still authenticate.

**SFTP Port**

The SFTP service uses a non-standard port number for connections to the appliance. This is to avoid conflicts with administrative SSH connections to port 22. By default, the SFTP port is 218 and must be specified on the SFTP client prior to connecting. For example, an OpenSolaris client using SFTP, would connect with the following command:

```
manta# sftp -o Port 218 root@guppy
```

**SFTP Logs**

<table>
<thead>
<tr>
<th>Log</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network-sftp:default</td>
<td>Logs SFTP service events</td>
</tr>
</tbody>
</table>

▼ **Allowing SFTP Access to a Share using the BUI**

1. Go to Configuration->Services
2. Check that the SFTP service is enabled and online. If not, enable the service.
3. Select or add a share in the Shares screen.
4. Go to the "Protocols" section, and check that SFTP access is enabled. This is also where the mode of access (read/read+write) can be set.

▼ **Configuring SFTP for Remote Access using the CLI**

1. Create a local user or network user (LDAP or NIS) with an appropriate administrator role. (See “Configuring Users” on page 124).
2. Generate an SSH authentication key by entering the command `ssh-keygen -t dsa` on the Solaris host/client.

3. Enter a file name in which to store the key.

4. Enter a passphrase if required, or leave this field blank to log on directly to the SFTP share. The location is displayed for the key. The key looks similar to the following:

```
ssh-dss AAAAB3NzaC1kc3MAAACBAPMMsSh8UWk1NPf/
VJDDEo00AwT+s6fZxkCmmrgAmLftTX9izWk+ bsvNldOiXN/6EgkusLjo/
+UaEt5+704vMHCIRaq3AIxHLS5tVjeX3iCs+fDo0qwxZg3Brh8QBAaWk3
ywr2ou9I1tHh4v/HwEAHzZ5mVWXav0p03bgmxl0/
+AAAFAqDJxnmm52DfyEdOQMT+yrRVvzGwMQAAAIAhTP6Ey
+2gGFCkkvUofscoc4dpbqH8duE9P6Y88so+opuj52GkAdRUr2fRrdM9Cf3h4I0c8Bw9
bZI8zrCKBNWBUdZG56tsfLdIL6wS6gxKrmL2v7fSp9WYPsZGhOLfU29zW4n2WVCvHbGyFEOVe
+taq aq+AYJaWoHnjZL1/
LpQAIAAOLe8+uc3hDoCk3pAkYdg8b2rYtGOAZU4py0rq24DGPeVHdshjbe4p
WDM70uYqGOCOPYoKeoMNjpczRX5qjI+BfoUY4sH24WWwsKt8X9PUIa0WT
+7axEqg2N6YelaTJ95J vMaj6E7HkAlra2Sj2H/LSDktL42UL+j1Wx5A== username sunray
```

5. Go to Configuration > Services > SFTP. Under Keys, click the plus (+) sign.

6. In the New Key window, select DSA.

7. Copy only the key portion (beginning with AAAA and ending with Wx5A== in the example above) and paste into the Key field. Enter the user name and add a comment as a reminder.
   
   **Note:** The key should not contain any white spaces.

8. Go to Shares > Shares and click the plus (+) sign to create a filesystem.

9. In the Create Filesystem window, enter the filesystem name (for example, sftp), change the permissions to Read/Write for the share, and click Apply.

10. Click the pencil icon to set up the share properties. (See “Working with Shares”.)

11. To access the share, use the `sftp` command as shown in these examples:

   ```
sftp -o "port=218" <username> 10.x.x.151:/export/sftp
   
   Connecting to 10.x.x.151...
   
   Changing to: /export/sftp
   
   sftp>
   
   Example with -v option:
   ```
Working with SRP

When you configure a LUN on the appliance you can export that volume over a SCSI Remote Protocol (SRP) target. The SRP service allows initiators to access targets using the SRP protocol.

For information on SRP targets and initiators, see “Configuring Storage Area Network” on page 99.

Working with TFTP

Trivial File Transfer Protocol (TFTP) is a simple protocol to transfer files. TFTP is designed to be small and easy to implement, therefore, lacks most of the features of a regular FTP. TFTP only reads and writes files (or mail) from/to a remote server. It cannot list directories, and currently has no provisions for user authentication.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Root Directory</td>
<td>The TFTP login location. The default is &quot;/export&quot; and points to the top of the shares hierarchy. All users will be logged into this location after successfully authenticating with the TFTP service</td>
</tr>
</tbody>
</table>

Allowing TFTP Access to a Share using the BUI

1. Go to Configuration->Services
2. Check that the TFTP service is enabled and online. If not, enable the service.
3. Select or add a share in the Shares screen.
4. Go to the "Protocols" section, and check that TFTP access is enabled. This is also where the mode of access (read/read+write) can be set.
Working with Virus Scan

The Virus Scan service will scan for viruses at the filesystem level. When a file is accessed from any protocol, the Virus Scan service will first scan the file, and both deny access and quarantine the file if a virus is found. Once a file has been scanned with the latest virus definitions, it is not rescanned until it is next modified. Files accessed by NFS clients that have cached file data or been delegated read privileges by the NFSv4 server may not be immediately quarantined.

To work with virus scan, see the following sections:
- “Virus Scan Properties” on page 220
- “Virus Scan File Extensions” on page 220
- “Scanning Engines” on page 221
- “Virus Scan Logs” on page 222
- “Configuring Virus Scanning for a Share using the BUI” on page 222

Virus Scan Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum file size to scan</td>
<td>Files larger than this size will not be scanned, to avoid significant performance penalties. These large files are unlikely to be executable themselves (such as database files), and so are less likely to pose a risk to vulnerable clients. The default value is 1GB.</td>
</tr>
<tr>
<td>Allow access to files that exceed maximum file size</td>
<td>Enabled by default, this allows access to files larger than the maximum scan size (which are therefore unscanned prior to being returned to clients). Administrators at a site with more stringent security requirements may elect to disable this option and increase the maximum file size, so that all accessible files are known to be scanned for viruses.</td>
</tr>
</tbody>
</table>

Virus Scan File Extensions

This section describes how to control which files are scanned. The default value, " * ", causes all files to be scanned. Scanning all files may impact performance so you can designate a subset of files to scan.

For example, to scan only high-risk files, including zip files, but not files with names that match the pattern "data-archive*.zip", you could configure the following settings:
### Working with Virus Scan

#### TABLE 77  Virus Scan File Extensions

<table>
<thead>
<tr>
<th>Action</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan</td>
<td>exe</td>
</tr>
<tr>
<td>Scan</td>
<td>com</td>
</tr>
<tr>
<td>Scan</td>
<td>bat</td>
</tr>
<tr>
<td>Scan</td>
<td>doc</td>
</tr>
<tr>
<td>Scan</td>
<td>zip</td>
</tr>
<tr>
<td>Don't Scan</td>
<td>data-archive*.zip</td>
</tr>
<tr>
<td>Don't Scan</td>
<td>*</td>
</tr>
</tbody>
</table>

**Note** - You must use “Don’t Scan *” to exclude all other file types not explicitly included in the scan list. A file named "file.name.exe.bat.jpg123" would NOT be scanned, as only the "jpg123" portion of the name, the extension, would be compared against the rules.

Do NOT use exclude settings before include settings. For example, do not use a "Don’t Scan *" setting before include settings since that would exclude all file types that come after it. The following example would not scan any files:

#### TABLE 78  Virus Scan Actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Don't Scan</td>
<td>*</td>
</tr>
<tr>
<td>Scan</td>
<td>exe</td>
</tr>
<tr>
<td>Scan</td>
<td>com</td>
</tr>
<tr>
<td>Scan</td>
<td>bat</td>
</tr>
<tr>
<td>Scan</td>
<td>doc</td>
</tr>
<tr>
<td>Scan</td>
<td>zip</td>
</tr>
<tr>
<td>Don't Scan</td>
<td>data-archive*.zip</td>
</tr>
</tbody>
</table>

### Scanning Engines

In this section, specify which scanning engines to use. A scanning engine is an external third-party virus scanning server which the appliance contacts using ICAP (Internet Content Adaptation Protocol, RFC 3507) to have files scanned.

#### TABLE 79  Scanning Engines Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable</td>
<td>Use this scan engine</td>
</tr>
<tr>
<td>Host</td>
<td>Hostname or IP address of the scan engine server</td>
</tr>
</tbody>
</table>
### Configuring Virus Scanning for a Share using the BUI

1. Go to Configuration>Services>Virus Scan.
2. Set desired properties.
3. Apply/commit the configuration.
4. Go to Shares.
5. Edit a filesystem or a project.
6. Select the "General" tab.
7. Enable the "Virus scan" option.

### Working with NIS

Network Information Service (NIS) is a name service for centralized management. The appliance can act as a NIS client for users and groups, so that:

- NIS users can login to the FTP and HTTP services.
- NIS users can be granted privileges for appliance administration. The appliance supplements NIS information with its own privilege settings.

**Note** - UIDs and GIDs from 0-99 inclusive are reserved by the operating system vendor for use in future applications. Their use by end system users or vendors of layered products is not supported and may cause security related issues with future applications.
To work with NIS, see the following sections:
- “NIS Properties” on page 223
- “NIS Logs” on page 223
- “Adding an Appliance Administrator from NIS using the BUI” on page 223

### NIS Properties

**TABLE 81  NIS Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>NIS domain to use</td>
</tr>
<tr>
<td>Server(s): Search using broadcast</td>
<td>The appliance will send a NIS broadcast to locate NIS servers for that domain</td>
</tr>
<tr>
<td>Server(s): Use listed servers</td>
<td>NIS server hostnames or IP addresses</td>
</tr>
</tbody>
</table>

The appliance will connect to the first NIS server listed or found using broadcast, and switch to the next if it stops responding.

### NIS Logs

**TABLE 82  NIS Logs**

<table>
<thead>
<tr>
<th>Log</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network-nis-client:default</td>
<td>NIS client service log</td>
</tr>
<tr>
<td>appliance-kit-nsswitch:default</td>
<td>Log of the appliance name service, through which NIS queries are made</td>
</tr>
<tr>
<td>system-identity:domain</td>
<td>Log of the appliance domainname configurator</td>
</tr>
</tbody>
</table>

### Adding an Appliance Administrator from NIS using the BUI

If you have an existing user in NIS who would like to login using their NIS credentials and administer the appliance:

**Note** - If both NIS and LDAP are configured on the appliance and the services return different information for a particular item, the appliance will use the data provided by NIS.
1. Go to Configuration>Services>NIS.
2. Set the NIS domain and server properties.
3. Apply/commit the configuration.
4. Go to Configuration>Users
5. Add user with type "directory"
6. Set username to their NIS username.
7. Continue with the instructions in “Configuring Users” on page 124 for adding authorizations to this user.

Working with LDAP

LDAP (Lightweight Directory Access Protocol) is a directory service for centralizing management of users, groups, hostnames and other resources (called objects). This service on the appliance acts as an LDAP client so that:
- LDAP users can log in to the FTP and HTTP services.
- LDAP user names (instead of numerical ids) can be used to configure root directory ACLs on a share.
- LDAP users can be granted privileges for appliance administration. The appliance supplements LDAP information with its own privilege settings.
- The LDAP server's certificate can be self-signed.
- You cannot supply a list of trusted CA certificates; each certificate must be individually accepted by the appliance administrator.
- When an LDAP server's certificate expires, you must delete the server from the list and then re-add it to accept its new certificate.

**Note** - UIDs from 0-99 inclusive are reserved by the operating system vendor for use in future applications. Their use by end system users or vendors of layered products is not supported and can cause security issues with other applications.

To work with LDAP, see the following sections:
- “LDAP Properties” on page 225
- “LDAP Custom Mappings” on page 226
- “LDAP Logs” on page 227
- “Adding an Appliance Administrator using the BUI” on page 227
LDAP Properties

For the appropriate settings for your environment, consult your LDAP server administrator.

- Protect LDAP traffic with SSL/TLS - Toggles TLS (Transport Layer Security, the descendant of SSL) to establish secure connections to the LDAP server.
- Base search DN - Supplies the distinguished name of the base object which is the starting point for directory searches.
- Search scope - Defines which objects in the LDAP directory are searched, relative to the base object. Search results can be limited only to objects directly beneath the base search object (one-level) or they can include any object beneath the base search object (subtree). The default is one-level.
- Authentication method - Method used to authenticate the appliance to the LDAP server. The appliance supports Simple (RFC 4513), SASL/DIGEST-MD5, and SASL/GSSAPI authentication. If the Simple authentication method is used, SSL/TLS should be enabled so the user's DN and password are not sent in plain text. When using the SASL/GSSAPI authentication method, only the self bind credential level is available.
- Bind credential level - Credentials used to authenticate the appliance to the LDAP server.
  - Anonymous gives the appliance access only to data that is available to everyone.
  - Proxy directs the service to bind via a specified account.
  - Proxy DN - Distinguished name of account used for proxy authentication.
  - Proxy Password - Password for account used for proxy authentication.
  - Self - Self authenticates the appliance using the user's identity and credentials. Self authentication can only be used with the SASL/GSSAPI authentication method.
- Schema definition - Schema used by the appliance. This property lets administrators override the default search descriptor, attribute mappings, and object class mappings for users, groups, and netgroups. For more information, see “Working with LDAP” on page 224.
- Servers - List of LDAP servers to use. If only one server is specified, the appliance uses only that server and LDAP services are unavailable if that server fails. If multiple servers are specified, any functioning server can be used at any time without preference. If any server fails, another server in the list is used. LDAP services remain available unless all specified servers fail.

Setting Properties with Multiple Attribute Value Pairs in the CLI

In the CLI, to set property values that have multiple attribute value pairs with equal signs (=), surround each attribute value pair with double quotation marks. For example:

```
zfssa:configuration services ldap> set user_mapattr="uid=uid",
  "uidnumber=uidNumber","gidnumber=gidNumber",
  "gecos=displayName","description=distinguishedName",
```
Working with LDAP

“homedirectory=unixHomeDirectory”

LDAP Custom Mappings

To look up users and groups in the LDAP directory, the appliance uses a search descriptor and must know which object classes correspond to users and groups and which attributes correspond to the properties needed. By default, the appliance uses object classes specified by RFC 2307 (posixAccount and posixGroup) and the default search descriptors shown in the following list, but this can be customized for different environments. The base search DN used in the examples below is dc=example,dc=com:

**TABLE 83 LDAP Custom Mappings**

<table>
<thead>
<tr>
<th>Search descriptor</th>
<th>Default value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>users</td>
<td>ou=people, base search DN</td>
<td>ou=people,dc=example,dc=com</td>
</tr>
<tr>
<td>groups</td>
<td>ou=group, base search DN</td>
<td>ou=group,dc=example,dc=com</td>
</tr>
<tr>
<td>netgroups</td>
<td>ou=netgroup, base search DN</td>
<td>ou=netgroup,dc=example,dc=com</td>
</tr>
</tbody>
</table>

The search descriptor, object classes, and attributes used can be customized using the Schema definition property. To override the default search descriptor, enter the entire DN you wish to use. The appliance will use this value unmodified, and will ignore the values of the Base search DN and Search scope properties. To override user, group, and netgroup attributes and objects, choose the appropriate tab (“Users”, “Groups”, or “Netgroups”) and specify mappings using the default = new syntax, where default is the default value and new is the value you want to use. For examples:

- To use unixaccount instead of posixAccount as the user object class, enter posixAccount = unixaccount in Object class mappings on the Users tab.
- To use employeenumber instead of uid as the attribute for user objects, enter uid = employeenumber in Attribute mappings on the Users tab.
- To use unixgroup instead of posixGroup as the group object class, type posixGroup = unixgroup in Object class mappings on the Groups tab.
- To use groupaccount instead of cn as the attribute for group objects, enter cn = groupaccount in Attribute mappings on the Groups tab.

The following is a list of object classes and attributes that you might want to map:

- Classes:
  - * posixAccount
  - * posixGroup
  - * shadowAccount
- Attributes - Users:
  - * uid
  - * uidNumber
■ * gidNumber
■ * gecos
■ * homeDirectory
■ * loginShell
■ * userPassword
■ Attributes - Groups:
■ * uid
■ * memberUid
■ * cn
■ * userPassword
■ * gidNumber
■ * member
■ * uniqueMember
■ * memberOf
■ * isMemberOf

LDAP Logs

The following is an example log.

<table>
<thead>
<tr>
<th>Log</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>appliance-kit-nsswitch:default</td>
<td>Log of the appliance name service, through which LDAP queries are made</td>
</tr>
</tbody>
</table>

Adding an Appliance Administrator using the BUI

To let an existing LDAP user log in using LDAP credentials and administer the appliance, use the following procedure:

**Note** - If both NIS and LDAP are configured on the appliance and the services return different information for a particular item, the appliance will use the data provided by NIS.

1. **On the Configuration>Services>LDAP page**, enter the properties that you want to use. For information about the available properties, see “Working with LDAP” on page 224.

2. **To apply properties you selected**, click Apply or click Revert to start over.
3. To add LDAP servers, in the Servers section click the add icon. For information about servers, see the Servers section in “Working with LDAP” on page 224.

4. To configure the LDAP server, in the New LDAP Server box, enter the LDAP server Address and select the LDAP Certificate source that you want to use. For the Certificate source, selecting Server searches the current server and retrieves the certificate (in an insecure manner) and uses it in the future to validate the certificate presented later.

5. On the Configuration>Users page, add users as needed using LDAP usernames. For information about adding users, see “Configuring Users” on page 124.

Working with Active Directory

The Active Directory service provides access to a Microsoft Active Directory database, which stores information about users, groups, shares, and other shared objects. This service has two modes: domain and workgroup mode, which dictate how SMB users are authenticated. When operating in domain mode, SMB clients are authenticated through the AD domain controller. In workgroup mode, SMB clients are authenticated locally as local users. See “Users” on page 124 for more information on local users.

To work with active directory, see the following sections:
- “Active Directory Join Domain” on page 228
- “Active Directory Join Workgroup” on page 229
- “Active Directory Domains and Workgroups” on page 229
- “Active Directory LDAP Signing” on page 230
- “Active Directory Windows Server Support” on page 230
- “Joining an AD Domain using the BUI” on page 231
- “Joining an AD Workgroup using the BUI” on page 231
- “Configuring Active Directory using the CLI” on page 231

Active Directory Join Domain

If an account does not already exist in Active Directory by default, a machine trust account for the system is automatically created in the default container for computer accounts (cn=Computers) as part of the domain join operation. The following users are allowed to perform domain join:
- Domain administrator - Can join any number of systems to the domain with machine trust accounts placed in any containers.
- **Delegated administrator with authority over one or more Organizational Units** - Can join any number of systems to a domain with machine account location designated in the Organizational Units they are responsible for.

- **Normal user with machine accounts pre-staged by administrator** - Can join a system to the domain as pre-authorized by an administrator.

- **Normal user** - Normally authorized to join a limited number of systems.

The following properties for joining an Active Directory domain are available:

- **Active Directory Domain** - The fully-qualified name or NetBIOS name of an Active Directory domain

- **User** - An AD user who has credentials to create a computer account in Active Directory

- **Password** - The administrative user's password

- **Additional DNS Search Path** - When this optional property is specified, DNS queries are resolved against this domain, in addition to the primary DNS domain and the Active Directory domain

- **Organizational Unit** - Specifies an alternative organizational unit in which the system's machine trust account will be created. The organizational unit is specified as a comma-separated list of one or more name-value pairs using the domain-relative distinguished name (DN) format, for example, ou=innerOU,ou=outerOU.

- **Use Pre-created Account** - If the system's account exists and the specified Organizational Unit is not the one that the account is in, use the pre-created account.

**Active Directory Join Workgroup**

The configurable property for joining a workgroup is Windows Workgroup.

**Active Directory Domains and Workgroups**

Instead of enabling and disabling the service directly, the service is modified by joining a domain or a workgroup. Joining a domain involves creating an account for the appliance in the given Active Directory domain. The account name can be a maximum of 15 characters, and must be unique to other names registered within the Active Directory domain. Otherwise, conflicts may occur with similarly named appliances and cause issues with functionality. After the computer account has been established, the appliance can securely query the database for information about users, groups, and shares.

Joining a workgroup implicitly leaves an Active Directory domain, and SMB clients who are stored in the Active Directory database will be unable to connect to shares.

If a Kerberos realm is configured to support Kerberized NFS, the system cannot be configured to join an Active Directory domain.
Active Directory LDAP Signing

There is no configuration option for LDAP signing, as that option is negotiated automatically when communicating with a domain controller. LDAP signing operates on communication between the storage appliance and the domain controller, whereas SMB signing operations on communication between SMB clients and the storage appliance.

Active Directory Windows Server Support

Windows Server 2012 is fully supported in software version 2011.1.5 and later.

<table>
<thead>
<tr>
<th>Windows Version</th>
<th>Supported Software Versions</th>
<th>Workarounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2003</td>
<td>All</td>
<td>None</td>
</tr>
<tr>
<td>Windows Server 2008 SP1</td>
<td>2009.Q2 3.1 and earlier</td>
<td>Apply hotfix for KB957441 as needed, see Section B.</td>
</tr>
<tr>
<td></td>
<td>2009.Q2 4.0 - 2011.1.1</td>
<td>Must apply hotfix for KB951191 and apply hotfix for KB957441 as needed, see Sections A and B.</td>
</tr>
<tr>
<td></td>
<td>2011.1.2 and later</td>
<td>Must apply hotfix for KB951191, see Section A.</td>
</tr>
<tr>
<td>Windows Server 2008 SP2</td>
<td>2009.Q2 4.0 - 2011.1.1</td>
<td>See Section C.</td>
</tr>
<tr>
<td></td>
<td>2011.1.2 and later</td>
<td>None</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>2009.Q2 4.0 - 2011.1.1</td>
<td>See Section C.</td>
</tr>
<tr>
<td></td>
<td>2011.1.2 and later</td>
<td>None</td>
</tr>
</tbody>
</table>

Active Directory Windows Server 2008 Support Section A: Kerberos issue (KB951191)

- If you upgrade to 2009.Q2.4.0 or later and your Windows 2008 domain controller is running Windows Server 2008 SP2 or R2, no action is required.
- If you upgrade to 2009.Q2.4.0 or later and your Windows 2008 domain controller is running Windows Server 2008 SP1, you must apply the hotfix described in KB951191 or install Windows 2008 SP2.

Active Directory Windows Server 2008 Support Section B: NTLMv2 issue (KB957441)

- The following applies only if your appliance is running a software version prior to 2011.1.2:
- If your Domain Controller is running Windows Server 2008 SP1 you should also apply the hotfix for [http://support.microsoft.com/kb/957441/](http://support.microsoft.com/kb/957441/) which resolves an NTLMv2 issue that prevents the appliance from joining the domain with its default LMCompatibilityLevel setting.
If the LMCompatibilityLevel on the Windows 2008 SP1 domain controller is set to 5, this hot fix must be installed. After applying the hotfix you must create and set a new registry key as described in KB957441.

If you upgrade to 2011.1.2 or later, you do not need the hotfix mentioned above.

Active Directory Windows Server 2008 Support Section C: Note on NTLMv2

The following applies only if your appliance is running a software version prior to 2011.1.2: If your Domain Controller is running Windows Server 2008 SP2 or R2 you do not need to apply the hotfix but you must apply the registry setting as described in KB957441.

If you upgrade to 2011.1.2 or later, no action is required.

▼ Joining an AD Domain using the BUI

1. Configure an Active Directory site in the SMB context. (optional)
2. Configure a preferred domain controller in the SMB context. (optional)
3. Enable NTP, or ensure that the clocks of the appliance and domain controller are synchronized to within five minutes.
4. Ensure that your DNS infrastructure correctly delegates to the Active Directory domain, or add your domain controller’s IP address as an additional name server in the DNS context.
5. Configure the Active Directory domain, administrative user, and administrative password.
6. Apply/commit the configuration.

▼ Joining an AD Workgroup using the BUI

1. Configure the workgroup name.
2. Apply/commit the configuration.

▼ Configuring Active Directory using the CLI

1. View an existing configuration.
2. Observe that the appliance is currently operating in the domain "eng.fishworks.com". Following is an example of leaving that domain and joining a workgroup.

```
twofish:configuration services ad> workgroup
```
```
twofish:configuration services ad workgroup> set workgroup=WORKGROUP
```
```
twofish:configuration services ad workgroup> commit
```
```
twofish:configuration services ad workgroup> done
```
```
twofish:configuration services ad> show
```
```
<status> = disabled
mode = workgroup
workgroup = WORKGROUP
```

3. Following is an example of configuring the site and preferred domain controller in preparation for joining another domain.

```
twofish:configuration services ad> done
```
```
twofish:> configuration services smb
```
```
twofish:configuration services smb> set ads_site=sf
```
```
twofish:configuration services smb> set pdc=192.168.3.21
```
```
twofish:configuration services smb> commit
```
```
twofish:configuration services smb> show
```
```
<status> = online
lauth_level = 4
   pdc = 192.168.3.21
ads_site = sf
```
```
twofish:configuration services smb> done
```

4. Following is an example of joining the new domain after the properties are configured. When joining an AD domain, you must set the user and password each time you commit the node.

```
twofish:> configuration services ad
```
```
twofish:configuration services ad> domain
```
Working with Identity Mapping

The identity mapping service manages Windows and Unix user identities simultaneously by using both traditional Unix UIDs (and GIDs) and Windows SIDs.

The identity mapping service creates and maintains a database of mappings between SIDs, UIDs, and GIDs. Three different mapping approaches are available; if mappings are available for a given identity, the service creates an ephemeral mapping. The following mapping modes are available:

- **Identity Mapping Rule-based Mapping** - The Rule-based mapping approach involves creating various rules which map identities by name. These rules establish equivalences between Windows identities and Unix identities.

- **Identity Mapping Directory-based Mapping** - Directory-based mapping involves annotating an LDAP or Active Directory object with information about how the identity maps to an equivalent identity on the opposite platform. The following attributes must be assigned when using directory-based mapping:
  - AD Attribute - Unix User Name - The name in the AD database of the equivalent Unix user name
  - AD Attribute - Unix Group Name - The name in the AD database of the equivalent Unix group name
  - Native LDAP Attribute - Windows User Name - The name in the LDAP database of the equivalent Windows identity

- **Identity Mapping IDMU** - Microsoft offers a feature called "Identity Management for Unix", or IDMU. This software is available for Windows Server 2003, and is bundled with Windows Server 2003 R2 and later. This feature is part of what was called "Services For Unix" in its unbundled form. The primary use of IDMU is to support Windows as a NIS/NFS server. IDMU adds a "UNIX Attributes" panel to the Active Directory Users and Computers user interface that lets the administrator specify a number of UNIX-related parameters: UID, GID, login shell, home directory, and similar for groups. These parameters are made available through AD through a schema similar to (but not the same as) RFC2307, and through the NIS service. When the IDMU mapping mode is selected, the identity mapping service consumes these Unix attributes to establish mappings between Windows and Unix identities. This approach is very similar to directory-based
mapping, only the identity mapping service queries the property schema established by the IDMU software instead of allowing a custom schema. When this approach is used, no other directory-based mapping may take place.

To work with identity mapping, see the following sections:
- “Identity Mapping Rules” on page 234
- “Mapping Rule Directional Symbols” on page 234
- “Identity Mapping Best Practices” on page 235
- “Identity Mapping Concepts” on page 235
- “Configuring Identity Mapping using the BUI” on page 237
- “Viewing or Flushing Mappings using the BUI” on page 238

Identity Mapping Rules

This page lets you create mappings using the following properties:
- **Mapping Type** - Allows or denies credentials. For more information, see “Deny Mappings” on page 234.
- **Mapping Direction** - The mapping direction. A mapping may map credentials in both directions, only from Windows to Unix, or only from Unix to Windows. For more information, see “Mapping Rule Directional Symbols” on page 234.
- **Windows Domain** - The Active Directory domain of the Windows identity.
- **Windows Identity** - The name of the Windows identity.
- **Unix Identity** - The name of the Unix identity.
- **Unix Identity Type** - The type of the Unix identity, either a user or a group.

Deny mapping rules prevent users from obtaining any mapping, including an ephemeral ID, from the identity mapping service. You can create domain-wide or user-specific deny mappings for Windows users and for Unix users. For example, you can create a mapping to deny access to SMB shares for all Unix users in the group “guest”. You cannot create deny mappings that conflict with other mappings.

Mapping Rule Directional Symbols

After creating a name-based mapping, the following symbols indicate the semantics of each rule.
- \[align="center|\leftarrow\rightarrow\] - Maps Windows identity to Unix identity, and Unix identity to Windows identity
- \[align="center|\leftarrow\] - Maps Windows identity to Unix identity
- \[align="center|\rightarrow\] - Maps Unix identity to Windows identity
- Prevents Windows identity from obtaining credentials
- Prevents Unix identity from obtaining credentials

If an icon is gray instead of black (◻, ◼, ◽, ◦, ◹), that rule matches a Unix identity which cannot be resolved.

Identity Mapping Best Practices

- Configuring fine-grained identity mapping rules only applies when you want to have the same user access a common set of files as both an NFS and SMB client. If NFS and SMB clients are accessing disjoint filesystems, there's no need to configure any identity mapping rules.
- Reconfiguring the identity mapping service has no effect on active SMB sessions. Connected users remain connected, and their previous name mapping is available for authorizing access to additional shares for up to 10 minutes. To prevent unauthorized access you must configure the mappings before you export shares.
- The security that your identity mappings provide is only as good as their synchronization with your directory services. For example, if you create a name-based mapping that denies access to a particular user, and the user's name changes, the mapping no longer denies access to that user.
- You can only have one bidirectional mapping for each Windows domain that maps all users in the Windows domain to all Unix identities. If you want to create multiple domain-wide rules, be sure to specify that those rules map only from Windows to Unix.
- Use the IDMU mapping mode instead of directory-based mapping whenever possible.

Identity Mapping Concepts

The SMB service uses the identity mapping service to associate Windows and Unix identities. When the SMB service authenticates a user, it uses the identity mapping service to map the user's Windows identity to the appropriate Unix identity. If no Unix identity exists for a Windows user, the service generates a temporary identity using an ephemeral UID and GID. These mappings allow a share to be exported and accessed concurrently by SMB and NFS clients. By associating Windows and Unix identities, an NFS and SMB client can share the same identity, thereby allowing access to the same set of files.

In the Windows operating system, an access token contains the security information for a login session and identifies the user, the user's groups, and the user's privileges. Administrators define Windows users and groups in a Workgroup, or in a SAM database, which is managed on an Active Directory domain controller. Each user and group has a SID. An SID uniquely identifies a user or group both within a host and a local domain, and across all possible Windows domains.
Unix creates user credentials based on user authentication and file permissions. Administrators define Unix users and groups in local password and group files or in a name or directory service, such as NIS and LDAP. Each Unix user and group has a UID and a GID. Typically, the UID or GID uniquely identifies a user or group within a single Unix domain. However, these values are not unique across domains.

Windows names are case-insensitive and Unix names are case-sensitive. The user names JSMITH, Jsmith, and jsmith are equivalent names in Windows, but they are three distinct names in Unix. Case sensitivity affects name mappings differently depending on the direction of the mapping.

- For a Windows-to-Unix mapping to produce a match, the case of the Windows username must match the case of the Unix user name. For example, only Windows user name "jsmith" matches Unix user name "jsmith". Windows user name "Jsmith" does not match.

- An exception to the case matching requirement for Windows-to-Unix mappings occurs when the mapping uses the wildcard character, "*" to map multiple user names. If the identity mapping service encounters a mapping that maps Windows user *@some.domain to Unix user "*", it first searches for a Unix name that matches the Windows name as-is. If it does not find a match, the service converts the entire Windows name to lower case and searches again for a matching Unix name. For example, the windows user name "JSmith@some.domain" maps to Unix user name "jsmith". If, after lowering the case of the Windows user name, the service finds no match, the user does not obtain a mapping. You can create a rule to match strings that differ only in case. For example, you can create a user-specific mapping to map the Windows user "JSmith@sun.com" to Unix user "jSmith". Otherwise, the service assigns an ephemeral ID to the Windows user.

- For a Unix-to-Windows mapping to produce a match, the case does not have to match. For example, Unix user name "jsmith" matches any Windows user name with the letters "JSMITH" regardless of case.

When the identity mapping service provides a name mapping, it stores the mapping for 10 minutes, at which point the mapping expires. Within its 10-minute life, a mapping is persistent across restarts of the identity mapping service. If the SMB server requests a mapping for the user after the mapping has expired, the service re-evaluates the mappings.

Changes to the mappings or to the name service directories do not affect existing connections within the 10-minute life of a mapping. The service evaluates mappings only when the client tries to connect to a share and there is no unexpired mapping.

A domain-wide mapping rule matches some or all of the names in a Windows domain to Unix names. The user names on both sides must match exactly (except for case sensitivity conflicts, which are subject to the rules discussed earlier). For example, you can create a bidirectional rule to match all Windows users in "myDomain.com" to Unix users with the same name, and vice-versa. For another example you can create a rule that maps all Windows users in "myDomain.com" in group "Engineering" to Unix users of the same name. You cannot create domain-wide mappings that conflict with other mappings.

If no name-based mapping rule applies for a particular user, that user will be given temporary credentials through an ephemeral mapping unless they are blocked by a deny mapping. When
a Windows user with an ephemeral Unix name creates a file on the system, Windows clients accessing the file using SMB see that the file is owned by that Windows identity. However, NFS clients see that the file is owned by “nobody”.

▼ Configuring Identity Mapping using the BUI

1. Ensure that you are joined to at least one active directory domain. For information about active directories, see “Active Directory” on page 228.

2. On the Configuration>Services>Identity Mapping>Properties page, select the Mapping mode you want to use. For information about mapping modes, see “Properties” on page 234.

3. If you select Directory-based Mapping, you must configure additional properties. For more information about these properties, see “Directory-based Mapping” on page 233.

4. To save your settings, click Apply or to start over click Revert.

5. To create a mappings, click Rules.

6. On the Rules page, click the add icon.

7. In the Add Mapping Rule box, enter the required information. For more information, see “Rules” on page 234.

8. To save your settings, click Add or click Cancel. When you create a mapping it appears in the Rules list.

Example 7 Example Adding Two Name-Based Rules using the CLI

This example creates a bi-directional name-based mapping between a Windows user and Unix user.

twofish:> configuration services idmap
twofish:configuration services idmap> create
twofish:configuration services idmap (uncommitted)> set windomain=eng.fishworks.com
twofish:configuration services idmap (uncommitted)> set winname=Bill
twofish:configuration services idmap (uncommitted)> set direction=bi
twofish:configuration services idmap (uncommitted)> set unixname=wdp
twofish:configuration services idmap (uncommitted)> set unixtype=user
twofish:configuration services idmap (uncommitted)> commit
Example 8  Example Creating a Deny Mapping

The next example creates a deny mapping to prevent all Windows users in a domain from obtaining credentials.

twofish:configuration services idmap> create
twofish:configuration services idmap (uncommitted)> list
Properties:
   windomain = (unset)
   winname = (unset)
   direction = (unset)
   unixname = (unset)
   unixtype = (unset)

twofish:configuration services idmap (uncommitted)> set
   windomain=guest.fishworks.com
   winname=*  
   direction=win2unix
   unixname=
   unixtype=user

twofish:configuration services idmap (uncommitted)> commit
twofish:configuration services idmap> list
MAPPING      WINDOWS ENTITY                    DIRECTION    UNIX ENTITY
idmap-000    Bill@eng.fishworks.com        (U) =>           wdp (U)

Viewing Mappings using the BUI

1. To view existing mappings, on the Configuration>Services>Identity Mapping>Mappings page, enter the required information.
2. Click Show. The mapping you designated appears.

Flush Mappings using the BUI

After a requested mapping has been provided, it is stored in the cache for 10 minutes and then expires. You can immediately expire a mapping by using the flush function, which expires all cached mappings:

1. Go to Configuration>Services>Identity Mapping>Mappings.
2. Click Flush. All cached mappings are expired.

**Working with DNS**

The DNS (Domain Name Service) client provides the ability to resolve IP addresses to hostnames and vice versa, and is always enabled on the appliance. Optionally, secondary hostname resolution via NIS and/or LDAP, if configured and enabled, may be requested for hostnames and addresses that cannot be resolved using DNS. Hostname resolution is used throughout the appliance user interfaces, including in Logs to indicate the location from which a user performed an auditable action and in Analytics to provide statistics on a per-client basis.

The configurable properties for the DNS client include a base domain name and a list of servers, specified by IP address. You must supply a domain name and at least one server address; the server must be capable of returning an NS (NameServer) record for the domain you specify, although it need not itself be authoritative for that domain.

To work with DNS, see the following sections:

- “DNS Properties” on page 239
- “Configuring DNS” on page 240
- “DNS Logs” on page 240
- “Active Directory and DNS” on page 240
- “Non-DNS Resolution” on page 240
- “DNS-Less Operation” on page 241

**DNS Properties**

**TABLE 86**  
DNS Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS Domain</td>
<td>Domain name to search first when performing partial hostname lookups</td>
</tr>
<tr>
<td>DNS Server(s)</td>
<td>One or more DNS servers. IP addresses must be used.</td>
</tr>
<tr>
<td>Allow IPv4 non-DNS resolution</td>
<td>IPv4 addresses may be resolved to hostnames, and hostnames to IPv4 addresses, using NIS and/or LDAP if configured and enabled.</td>
</tr>
<tr>
<td>Allow IPv6 non-DNS resolution</td>
<td>IPv4 and IPv6 addresses may be resolved to hostnames, and hostnames to IPv4 and IPv6 addresses, using NIS and/or LDAP if configured and enabled.</td>
</tr>
</tbody>
</table>

Changing services properties is documented in “Services BUI Page Icons and Buttons” on page 177 and “Configuring Services using the CLI” on page 178. The CLI property names are shorter versions of those listed above.
Configuring DNS

The CLI includes built-ins for `nslookup` and `getent hosts`, which can be used to test that hostname resolution is working:

```
caji:> nslookup deimos
192.168.1.109   deimos.sf.fishworks.com
caji:> getent hosts deimos
192.168.1.109   deimos.sf.fishworks.com
```

DNS Logs

<table>
<thead>
<tr>
<th>Log</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network-dns-client:default</td>
<td>Logs the DNS service events</td>
</tr>
</tbody>
</table>

Active Directory and DNS

If you plan to use Active Directory, the servers must be able to resolve hostname and server records in the Active Directory portion of the domain namespace. For example, if your appliance resides in the domain example.com and the Active Directory portion of the namespace is redmond.example.com, your nameservers must be able to reach an authoritative server for example.com, and they must provide delegation for the domain redmond.example.com to one or more Active Directory servers serving that domain. These are requirements imposed by Active Directory, not the appliance itself. If they are not satisfied, you will be unable to join an Active Directory domain.

Non-DNS Resolution

DNS is a standard, enterprise-grade, highly-scalable and reliable mechanism for mapping between hostnames and IP addresses. Use of working DNS servers is a best practice and will generally yield the best results. In some environments, there may be a subset of hosts that can be resolved only in NIS or LDAP maps. If this is the case in your environment, enable non-DNS host resolution and configure the appropriate directory service(s). If LDAP is used for host resolution, the hosts map must be located at the standard DN in your database: `ou=Hosts, (Base DN)`, and must use the standard schema. When this mode is used with NFS sharing by netgroups, it may be necessary for client systems to use the same hostname resolution mechanism configured on the appliance, or NFS sharing exceptions may not work correctly.
When non-DNS host resolution is enabled, DNS will still be used. Only if an address or hostname cannot be resolved using DNS will NIS (if enabled) and then LDAP (if enabled) be used to resolve the name or address. This can have confusing and seemingly inconsistent results. You can validate host resolution results using the `getent` CLI command described above.

Use of these options is strongly discouraged.

## DNS-Less Operation

DNS-less operation is not supported on the appliance and could cause undesirable results. Several features do not operate correctly without DNS, including but not limited to:

- Analytics will be unable to resolve client addresses to hostnames.
- The Active Directory feature will not function (you will be unable to join a domain).
- Use of SSL-protected LDAP will not work properly with certificates containing hostnames.
- Alert and threshold actions that involve sending e-mail can only be sent to mail servers on an attached subnet, and all addresses must be specified using the mail server's IP address.
- Some operations may take longer than normal due to hostname resolution timeouts.

## Working with Dynamic Routing

The RIP (Routing Information Protocol) is a distance-vector dynamic routing protocol that is used by the appliance to automatically configure optimal routes based on messages received from other RIP-enabled on-link hosts (typically routers). The appliance supports both RIPv1 and RIPv2 for IPv4, and RIPng for IPv6. Routes that are configured via these protocols are marked as type "dynamic" in the routing table. RIP and RIPng listen on UDP ports 520 and 521 respectively.

<table>
<thead>
<tr>
<th>Log</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network-routing-route:default</td>
<td>Logs RIP service events</td>
</tr>
<tr>
<td>network-routing-ripng:quagga</td>
<td>Logs RIPng service events</td>
</tr>
</tbody>
</table>

## Working with IPMP

IPMP (Internet Protocol Network Multipathing) allows multiple network interfaces to be grouped as one, for both improved network bandwidth and reliability (interface redundancy).
Working with NTP

Some properties can be configured in this section. For the configuration of network interfaces in IPMP groups, see “Network Configuration” on page 56.

<table>
<thead>
<tr>
<th>TABLE 89</th>
<th>IPMP Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>Failure detection latency</td>
<td>Time for IPMP to declare a network interface has failed, and to fail over its IP addresses</td>
</tr>
<tr>
<td>Enable fail-back</td>
<td>Allow the service to resume connections to a repaired interface</td>
</tr>
</tbody>
</table>

Changing services properties is documented in “Services BUI Page Icons and Buttons” on page 177 and “Configuring Services using the CLI” on page 178. The CLI property names are shorter versions of those listed above.

<table>
<thead>
<tr>
<th>TABLE 90</th>
<th>IPMP Logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log</td>
<td>Description</td>
</tr>
<tr>
<td>network-initial:default</td>
<td>Logs the network configuration process</td>
</tr>
</tbody>
</table>

Working with NTP

The Network Time Protocol (NTP) service can be used to keep the appliance clock accurate. This is important for recording accurate timestamps in the filesystem, and for protocol authentication. The appliance records times using the UTC timezone. The times that are displayed in the BUI use the timezone offset of your browser.

To the right of the BUI screen are times from both the appliance (Server Time) and your browser (Client Time). If the NTP service is not online, the “SYNC” button can be clicked to set the appliance time to match your client browser time.

If you are sharing filesystems using SMB, the client clocks must be synchronized to within five minutes of the appliance clock to avoid user authentication errors. One way to ensure clock synchronization is to configure the appliance and the SMB clients to use the same NTP server.

<table>
<thead>
<tr>
<th>TABLE 91</th>
<th>NTP Clock Synchronization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log</td>
<td>Description</td>
</tr>
<tr>
<td>network-ntp:default</td>
<td>Log for the NTP service</td>
</tr>
</tbody>
</table>

To work with NTP, see the following sections:
- “NTP Properties” on page 243
Working with NTP

- “Setting Clock Synchronization using the BUI” on page 244
- “Configuring NTP using the CLI” on page 244

NTP Properties

<table>
<thead>
<tr>
<th>TABLE 92</th>
<th>NTP Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>multicast address</td>
<td>Enter a multicast address here for an NTP server to be located automatically</td>
</tr>
<tr>
<td>NTP server(s)</td>
<td>Enter one or more NTP servers (and their corresponding authentication keys, if any) for the appliance to contact directly</td>
</tr>
<tr>
<td>NTP Authentication Keys</td>
<td>Enter one or more NTP authentication keys for the appliance to use when authenticating the validity of NTP servers. See the Authentication section below.</td>
</tr>
</tbody>
</table>

If an invalid configuration is entered, a warning message is displayed and the configuration is not committed. This will happen if:

- A multicast address is used but no NTP response is found.
- An NTP server address is used, but that server does not respond properly to NTP.

To prevent against NTP spoofing attacks from rogue servers, NTP has a private key encryption scheme whereby NTP servers are associated with a private key that is used by the client to verify their identity. These keys are not used to encrypt traffic, and they are not used to authenticate the client -- they are only used by the NTP client (that is, the appliance) to authenticate the NTP server. To associate a private key with an NTP server, the private key must first be specified. Each private key has a unique integer associated with it, along with a type and key. The type must be one of the following:

<table>
<thead>
<tr>
<th>TABLE 93</th>
<th>NTP Private Keys and Integers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>DES</td>
<td>A 64 bit hexadecimal number in DES format</td>
</tr>
<tr>
<td>NTP</td>
<td>A 64 bit hexadecimal number in NTP format</td>
</tr>
<tr>
<td>ASCII</td>
<td>A 1-to-8 character ASCII string</td>
</tr>
<tr>
<td>MD5</td>
<td>A 1-to-8 character ASCII string, using the MD5 authentication scheme.</td>
</tr>
</tbody>
</table>
After the keys have been specified, an NTP server can be associated with a particular private key. For a given key, all of the key number, key type and private key values must match between client and server for an NTP server to be authenticated.

### Setting Clock Synchronization using the BUI

This will set the appliance time to match the time of your browser.

1. Disable the NTP service.
2. Click the "SYNC" button.

### Configuring NTP using the CLI

1. **Under configuration services ntp, edit authorizations with the authkey command:**

   ```
   clownfish:configuration services ntp> authkey
   clownfish:configuration services ntp authkey>
   ```

2. **From this context, new keys can be added with the create command:**

   ```
   clownfish:configuration services ntp authkey> create
   clownfish:configuration services ntp authkey-000 (uncommitted)> get
   keyno = (unset)
   type = (unset)
   key = (unset)
   clownfish:configuration services ntp authkey-000 (uncommitted)> set keyno=1
   keyno = 1 (uncommitted)
   key = (unset)
   clownfish:configuration services ntp authkey-000 (uncommitted)> set type=A
   type = A (uncommitted)
   clownfish:configuration services ntp authkey-000 (uncommitted)> set key=coconuts
   key = ******** (uncommitted)
   clownfish:configuration services ntp authkey-000 (uncommitted)> commit
   ```

3. **To associate authentication keys with servers via the CLI, the serverkeys property should be set to a list of values in which each value is a key to be associated with the corresponding server in the servers property. If a server does not use authentication, the corresponding server key should be set to 0. For example, to use the key created above to authenticate the servers "gefilte" and "carp":**

   ```
   clownfish:configuration services ntp> set servers=gefilte,carp
   servers = gefilte,carp (uncommitted)
   ```
4. To associate authentication keys with servers via the CLI, the serverkeys property should be set to a list of values in which each value is a key to be associated with the corresponding server in the servers property. If a server does not use authentication, the corresponding server key should be set to 0. For example, to use the key created above to authenticate the servers "gefilte" and "carp":

```
cloonfish:configuration services ntp> set serverkeys=1,1
    serverkeys = 1,1 (uncommitted)
cloonfish:configuration services ntp> commit
cloonfish:configuration services ntp>
```

5. To authenticate the server "gefilte" with key 1, "carp" with key 2 and "dory" with key 3:

```
cloonfish:configuration services ntp> set servers=gefilte,carp
    servers = gefilte,carp (uncommitted)
cloonfish:configuration services ntp> set serverkeys=1,2
    serverkeys = 1,2 (uncommitted)
cloonfish:configuration services ntp> commit
cloonfish:configuration services ntp>
```

6. To authenticate the servers "gefilte" and "carp" with key 1, and to additionally have an unauthenticated NTP server "dory":

```
cloonfish:configuration services ntp> set servers=gefilte,carp,dory
    servers = gefilte,carp,dory (uncommitted)
cloonfish:configuration services ntp> set serverkeys=1,1,0
    serverkeys = 1,1,0 (uncommitted)
cloonfish:configuration services ntp> commit
cloonfish:configuration services ntp>
```

**Working with Phone Home**

The Phone Home service screen is used to manage the appliance registration as well as the Phone Home remote support service.

Registration connects your appliance with Oracle Auto Service Request (http://www.oracle.com/us/support/auto-service-request/index.html). Oracle ASR automatically opens Service Requests (SR) for specific problems reported by your appliance.
Registration also connects your appliance with My Oracle Support (MOS) to detect update notifications.

The Phone Home service communicates with Oracle support to provide:

- **Fault reporting** - The system reports active problems to Oracle for automated service response. Depending on the nature of the fault, a support case may be opened. Details of these events can be viewed in Problems. For more information, see “Problems” in “Oracle ZFS Storage Appliance Customer Service Manual”.

- **Heartbeats** - Daily heartbeat messages are sent to Oracle to indicate that the system is up and running. Oracle support may notify the technical contact for an account when one of the activated systems fails to send a heartbeat for too long.

- **System configuration** - Periodic messages are sent to Oracle describing current software and hardware versions and configuration as well as storage configuration. No user data or metadata is transmitted in these messages.

- **Support bundles** - The Phone Home service must be enabled before support bundles can be uploaded to Oracle Support. See “System” in “Oracle ZFS Storage Appliance Customer Service Manual” for more information.


You must register to use the Phone Home service.

You need a valid Oracle Single Sign-On account user name and password to use the fault reporting and heartbeat features of the Phone Home service. Go to [My Oracle Support](http://support.oracle.com) and click Register to create your account.

To work with Phone Home, see the following sections:

- “Phone Home Properties” on page 246
- “Registering the Appliance using the BUI” on page 247
- “Registering the Appliance using the CLI” on page 247
- “Changing Account Information using the BUI” on page 248

### Phone Home Properties

If the appliance is not directly connected to the Internet, you may need to configure an HTTP proxy through which the Phone Home service can communicate with Oracle. These proxy settings will also be used to upload support bundles. See “System” in “Oracle ZFS Storage Appliance Customer Service Manual” for more details on support bundles.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use proxy</td>
<td>Connect via a web proxy</td>
</tr>
</tbody>
</table>
Registering the Appliance using the BUI

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host/port</td>
<td>Web proxy hostname or IP address, and port</td>
</tr>
<tr>
<td>Username</td>
<td>Web proxy username</td>
</tr>
<tr>
<td>Password</td>
<td>Web proxy password</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last heartbeat sent at</td>
<td>Time last heartbeat was sent to Oracle support</td>
</tr>
</tbody>
</table>

If the Phone Home service is enabled before a valid Oracle Single Sign-On account has been entered, it will appear in the maintenance state. You must enter a valid Oracle Single Sign-On account to use the Phone Home service.

There is a log of Phone Home events in Maintenance > Logs > Phone Home.

▶ Registering the Appliance using the BUI

1. Enter your Oracle Single Sign-On Account user name and password. A privacy statement is displayed. It can be viewed at any time in both the BUI and CLI.
2. Commit your changes.

▶ Registering the Appliance using the CLI

1. Set soa_id and soa_password to the user name and password for your Oracle Single Sign-On Account, respectively.
2. Commit your changes.
Example 9  CLI Registration

    dory:> configuration services scrk
    dory:configuration services scrk> set soa_id=myuser
      soa_id = myuser (uncommitted)
    dory:configuration services scrk> set soa_password=mypass
      soa_password = ****** (uncommitted)
    dory:configuration services scrk> commit

▼ Changing Account Information using the BUI

1. Click 'Change account...' to change the Oracle Single Sign-On Account used by the appliance.
2. Commit your changes.
3. Use My Oracle Support to complete Auto Service Request (ASR) activation. Refer to "How To Manage and Approve Pending ASR Assets In My Oracle Support" (Doc ID 1329200.1)

Working with the RESTful API

The Oracle ZFS Storage Appliance RESTful API lets you manage the appliance using simple requests such as GET, PUT, POST, and DELETE HTTP against managed resource URL paths.

The appliance RESTful based architecture is defined as a layered client-server model. Advantages of this model mean that services can be transparently redirected through standard hubs, routers, and other network systems without client configuration. This architecture supports caching of information and is useful when many clients request the same static resources.

For complete Oracle ZFS Storage Appliance RESTful API documentation, see Oracle ZFS Storage Appliance Documentation.

Working with Service Tags

Service Tags are used to facilitate product inventory and support, by allowing the appliance to be queried for data such as:

- System serial number
- System type
- Software version numbers
You can register the service tags with Oracle support, allowing you to easily keep track of your Oracle equipment and also expedite service calls. The service tags are enabled by default.

<table>
<thead>
<tr>
<th>TABLE 96</th>
<th>UDP/TCP Port Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>Discovery Port</td>
<td>UDP port used for service tag discovery. Default is 6481</td>
</tr>
<tr>
<td>Listener Port</td>
<td>TCP port used to query service tag data. Default is 6481</td>
</tr>
</tbody>
</table>

**Working with SMTP**

The SMTP service sends all mail generated by the appliance, typically in response to alerts as configured on the Alerts screen. The SMTP service does not accept external mail - it only sends mail generated automatically by the appliance itself.

By default, the SMTP service uses DNS (MX records) to determine where to send mail. If DNS is not configured for the appliance's domain, or the destination domain for outgoing mail does not have DNS MX records setup properly, the appliance can be configured to forward all mail through an outgoing mail server, commonly called a smarthost.

<table>
<thead>
<tr>
<th>TABLE 97</th>
<th>SMTP Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>Send mail through smarthost</td>
<td>If enabled, all mail is sent through the specified outgoing mail server. Otherwise, DNS is used to determine where to send mail for a particular domain.</td>
</tr>
<tr>
<td>Smarthost hostname</td>
<td>Outgoing mail server hostname.</td>
</tr>
<tr>
<td>Allow customized from address</td>
<td>If enabled, the From address for email is set to the Custom from address property. It may be desirable to customize this if the default From address is being identified as spam, for example.</td>
</tr>
<tr>
<td>Custom from address</td>
<td>The From address to use for outbound email.</td>
</tr>
</tbody>
</table>

When changing properties, you can use Alerts to send a test email to verify that the properties are correct. A common reason for undelivered email is misconfigured DNS, which prevents the appliance from determining which mail server to deliver the mail to; as described earlier, a smarthost could be used if DNS cannot be configured.

<table>
<thead>
<tr>
<th>TABLE 98</th>
<th>SMTP Logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log</td>
<td>Description</td>
</tr>
<tr>
<td>network-smtp:sendmail</td>
<td>Logs the SMTP service events</td>
</tr>
<tr>
<td>mail</td>
<td>Log of SMTP activity (including mails sent)</td>
</tr>
</tbody>
</table>
Working with SNMP

The SNMP (Simple Network Management Protocol) service provides two different functions on the appliance:

- Appliance status information can be served by SNMP.
- Alerts can be configured to send SNMP traps. See “Configuring Alerts” on page 134.

SNMP versions v1, v2c, and v3 are available when this service is enabled. The appliance supports a maximum of 50 physical and logical network interfaces. More than 50 network interfaces could cause time outs for such commands as snmpwalk and snmpget. If you need more than 50 network interfaces, contact Oracle Support.

To work with SNMP, see the following sections:

- “SNMP Properties” on page 250
- “SNMP MIBs” on page 251
- “Sun FM MIB” on page 251
- “Sun AK MIB” on page 252
- “Configuring SNMP to Serve Appliance Status using the BUI” on page 252
- “Configuring SNMP to Send Traps using the BUI” on page 252

SNMP Properties

- Version: Toggles between v1/2c and v3.
- Community name: Toggles between public and user-input. If you select user-input, you must also enter a community name. If you select v3, this property is not available.
- Authorized network/subnet: Enter an appropriate IPv4 address and subnet (integers from 0-32). If you select v3, this property is not available.
- Appliance contact: Enter an appropriate appliance contact.
- Username/password: Enter a valid username (max 501 characters) and password (8-501 characters). If you select v1/2c, this property is not available.
- Authentication: Toggles between MD5 and SHA authentication algorithms. If you select v1/2c, this property is not available.
- Privacy: Toggles between None and DES encryption algorithm. If you select v1/2c, this property is not available.
- Engine ID: The EngineID value hashed by snmpd. If SNMP was not previously enabled, the label shows “0x000”.
- Trap destinations: Lets you add IPv4 addresses. Use the “+” and “-” buttons to add or remove addresses.

The SNMP service also provides the MIB-II location string. This property is sourced from the “System Identity” on page 233 configuration.
SNMP MIBs

If the SNMP services is online, authorized networks will have access to the following MIBs (Management Information Bases):

### TABLE 99  SNMP MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1.3.6.1.2.1.1</td>
<td>MIB-II system - generic system information, including hostname, contact and location</td>
</tr>
<tr>
<td>.1.3.6.1.2.1.2</td>
<td>MIB-II interfaces - network interface statistics</td>
</tr>
<tr>
<td>.1.3.6.1.2.1.4</td>
<td>MIB-II IP - Internet Protocol information, including IP addresses and route table</td>
</tr>
<tr>
<td>.1.3.6.1.4.1.42</td>
<td>Sun Enterprise MIB (SUN-MIB.mib.txt)</td>
</tr>
<tr>
<td>.1.3.6.1.4.1.42.2.195</td>
<td>Sun FM - fault management statistics (MIB file linked below)</td>
</tr>
<tr>
<td>.1.3.6.1.4.1.42.2.225</td>
<td>Sun AK - appliance information and statistics (MIB file linked below)</td>
</tr>
</tbody>
</table>

**Note** - Sun MIB files are available at [https://your IP address or host name]:215/docs/snmp/.

### Sun FM MIB

The Sun FM MIB (SUN-FM-MIB.mib) provides access to SUN Fault Manager information such as:
- Active problems on the system
- Fault Manager events
- Fault Manager configuration information

There are four main tables to read:

### TABLE 100  Sun FM MIBs

<table>
<thead>
<tr>
<th>OID</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1.3.6.1.4.1.42.2.195.1.1</td>
<td>Fault Management problems</td>
</tr>
<tr>
<td>.1.3.6.1.4.1.42.2.195.1.2</td>
<td>Fault Management fault events</td>
</tr>
<tr>
<td>.1.3.6.1.4.1.42.2.195.1.3</td>
<td>Fault Management module configuration</td>
</tr>
<tr>
<td>.1.3.6.1.4.1.42.2.195.1.5</td>
<td>Fault Management faulty resources</td>
</tr>
</tbody>
</table>

See the MIB file linked above for the full descriptions.
Sun AK MIB

The Sun AK MIB (SUN-AK-MIB.mib) provides the following information:

- Product description string and part number
- Appliance software version
- Appliance and chassis serial numbers
- Install, update and boot times
- Cluster state
- Share status (share name, size, used and available bytes)

There are three main tables to read:

<table>
<thead>
<tr>
<th>OID</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1.3.6.1.4.1.42.2.225.1.4</td>
<td>General appliance info</td>
</tr>
<tr>
<td>.1.3.6.1.4.1.42.2.225.1.5</td>
<td>Cluster status</td>
</tr>
<tr>
<td>.1.3.6.1.4.1.42.2.225.1.6</td>
<td>Share status</td>
</tr>
</tbody>
</table>

See the MIB file linked above for the full descriptions.

▼ Configuring SNMP to Serve Appliance Status using the BUI

1. Set the community name, authorized network and contact string.

2. If desired, set the trap destination to a remote SNMP host, else set this to 127.0.0.1.

3. Apply/commit the configuration.

4. Restart the service.

▼ Configuring SNMP to Send Traps using the BUI

1. Set the community name, contact string, and trap destination(s).
2. If desired, set the authorized network to allow SNMP clients, else set this to 127.0.0.1/8.

3. Apply/commit the configuration.

4. Restart the service.

5. You must configure alerts to send the traps you want to receive. For more information about alerts, see “Configuring Alerts” on page 134.

Working with Syslog

The Syslog Relay service provides two different functions on the appliance:

■ Alerts can be configured to send Syslog messages to one or more remote systems. See “Configuring Alerts” on page 134.
■ Services on the appliance that are syslog capable will have their syslog messages forwarded to remote systems.

A syslog message is a small event message transmitted from the appliance to one or more remote systems (or as we like to call it: intercontinental printf). The message contains the following elements:

■ A facility describing the type of system component that emitted the message
■ A severity describing the severity of the condition associated with the message
■ A timestamp describing the time of the associated event in UTC
■ A hostname describing the canonical name of the appliance
■ A tag describing the name of the system component that emitted the message. See below for details of the message format.
■ A message describing the event itself. See below for details of the message format.

Syslog receivers are provided with most operating systems, including Solaris and Linux. A number of third-party and open-source management software packages also support Syslog. Syslog receivers allow administrators to aggregate messages from a number of systems on to a single management system and incorporated into a single set of log files.

The Syslog Relay can be configured to use the "classic" output format described by RFC 3164, or the newer, versioned output format described by RFC 5424. Syslog messages are transmitted as UDP datagrams. Therefore they are subject to being dropped by the network, or may not be sent at all if the sending system is low on memory or the network is sufficiently congested. Administrators should therefore assume that in complex failure scenarios in a network some messages may be missing and were dropped.

To work with syslog, see the following sections:
Syslog Properties

**TABLE 102**   Syslog Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol Version</td>
<td>The version of the Syslog protocol to use, either Classic or Modern</td>
</tr>
<tr>
<td>Destinations</td>
<td>The list of destination IPv4 and IPv6 addresses to which messages are relayed.</td>
</tr>
</tbody>
</table>

**Classic Syslog: RFC 3164**

The Classic Syslog protocol includes the facility and level values encoded as a single integer priority, the timestamp, a hostname, a tag, and the message body.

The tag will be one of the tags described below.

The hostname will be the canonical name of the appliance as defined by the System Identity configuration. For more information, see “System Identity” on page 233.

**Updated Syslog: RFC 5424**

The Classic Syslog protocol includes the facility and level values encoded as a single integer priority, a version field (1), the timestamp, a hostname, an app-name, and the message body. Syslog messages relayed by the Sun Storage systems will set the RFC 5424 procid, msgid, and structured-data fields to the nil value (\(-\)) to indicate that these fields do not contain any data.

The app-name will be one of the tags described below.

The hostname will be the canonical name of the appliance as defined by the System Identity configuration. For more information, see “System Identity” on page 233.
SYSLOG Message Format

The Syslog protocol itself does not define the format of the message payload, leaving it up to the sender to include any kind of structured data or unstructured human-readable string that is appropriate. Sun Storage appliances use the syslog subsystem tag ak to indicate a structured, parseable message payload, described next. Other subsystem tags indicate arbitrary human-readable text, but administrators should consider these string forms unstable and subject to change without notice or removal in future releases of the Sun Storage software.

**TABLE 103** SYSLOG Message Formats

<table>
<thead>
<tr>
<th>Facility</th>
<th>Tag Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>daemon</td>
<td>ak</td>
<td>Generic tag for appliance subsystems. All alerts will be tagged ak, indicating a SUNW-MSG-ID follows.</td>
</tr>
</tbody>
</table>

SYSLOG Alert Message Format

If an alert is configured with the Send Syslog Message action, it will produce a syslog message payload containing localized text consisting of the following standard fields. Each field will be prefixed with the field name in CAPITAL letters followed by a colon and whitespace character.

**TABLE 104** SYSLOG Alert Message Formats

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNW-MSG-ID</td>
<td>The stable Sun Fault Message Identifier associated with the alert. Each system condition and fault diagnosis that produces an administrator alert is assigned a persistent, unique identifier in Sun’s Fault Message catalog. These identifiers can be easily read over the phone or scribbled down in your notebook, and link to a corresponding knowledge article found at sun.com/msg/.</td>
</tr>
<tr>
<td>TYPE</td>
<td>The type of condition. This will be one of the labels: Fault, indicating a hardware component or connector failure; Defect indicating a software defect or misconfiguration; Alert, indicating a condition not associated with a fault or defect, such as the completion of a backup activity or remote replication.</td>
</tr>
<tr>
<td>VER</td>
<td>The version of this encoding format itself. This description corresponds to version “1” of the SUNW-</td>
</tr>
</tbody>
</table>
### Field Name | Description
---|---
**MSG-ID format. If a "1" is present in the VER field, parsing code may assume that all of the subsequent fields will be present. Parsing code should be written to handle or ignore additional fields if a decimal integer greater than one is specified.**

**SEVERITY** | The severity of the condition associated with the problem that triggered the alert. The list of severities is shown below.

**EVENT-TIME** | The time corresponding to this event. The time will be in the form "Day Mon DD HH:MM:SS YYYY" in UTC. For example: Fri Aug 14 21:34:22 2009.

**PLATFORM** | The platform identifier for the appliance. This field is for Oracle Service use only.

**CSN** | The chassis serial number of the appliance.

**HOSTNAME** | The canonical name of the appliance as defined by the System Identity configuration. See “System Identity” on page 258.

**SOURCE** | The subsystem within the appliance software that emitted the event. This field is for Oracle Service use only.

**REV** | The internal revision of the subsystem. This field is for Oracle Service use only.

**EVENT-ID** | The Universally Unique Identifier (UUID) associated with this event. Oracle's Fault Management system associates a UUID with each alert and fault diagnosis such that administrators can gather and correlated multiple messages associated with a single condition, and detect duplicate messages. Oracle Service personnel can use the EVENT-ID to retrieve additional postmortem information associated with the problem that may help Oracle respond to the issue.

**DESC** | Description of the condition associated with the event.

**AUTO-RESPONSE** | The automated response to the problem, if any, by the Fault Management software included in the system. Automated responses include capabilities such as proactively offlining faulty disks, DRAM memory chips, and processor cores.

**REC-ACTION** | The recommended service action. This will include a brief summary of the recommended action, but administrators should consult the knowledge article and this documentation for information on the complete repair procedure.

The **SEVERITY** field will be set to one of the following values:

<table>
<thead>
<tr>
<th>Severity</th>
<th>Syslog Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minor</td>
<td>LOG_WARNING</td>
<td>A condition occurred that does not currently impair service, but the</td>
</tr>
</tbody>
</table>
### Example Configuring a Solaris Receiver using the CLI

Most operating systems include a syslog receiver, but some configuration steps may be required to turn it on. Consult the documentation for your operating system or management software for specific details of syslog receiver configuration.

Solaris includes a bundled syslogd(1M) that can act as a syslog receiver, but the remote receive capability is disabled by default. To enable Solaris to receive syslog traffic, use svccfg and svcadm to modify the syslog settings as follows:

```bash
# svccfg -s system/system-log setprop config/log_from_remote = true
# svcadm refresh system/system-log
```

Solaris syslogd only understands the Classic Syslog protocol. Refer to the Solaris syslog.conf(4) man page for information on how to configure filtering and logging of the received messages.

By default, Solaris syslogd records messages to /var/adm/messages and a test alert would be recorded as follows:

```
TYPE: alert, VER: 1, SEVERITY: Minor\nEVENT-TIME: Fri Aug 14 21:34:22 2009\nPLATFORM: i86pc, CSN: 12345678, HOSTNAME: poptart\nSOURCE: jsui.359, REV: 1.0\nEVENT-ID: 92dfeb39-6e15-e2d5-a7d9-dc3e221bece5\nDESC: A test alert has been posted.\nAUTO-RESPONSE: None.\nIMPACT: None.\nREC-ACTION: None.
```

### Example Configuring a Linux Receiver using the CLI

Most operating systems include a syslog receiver, but some configuration steps may be required to turn it on. Consult the documentation for your operating system or management software for specific details of syslog receiver configuration.

<table>
<thead>
<tr>
<th>Severity</th>
<th>Syslog Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major</td>
<td>LOG_ERR</td>
<td>A condition occurred that does impair service but not seriously.</td>
</tr>
<tr>
<td>Critical</td>
<td>LOG_CRIT</td>
<td>A condition occurred that seriously impairs service and requires immediate correction.</td>
</tr>
</tbody>
</table>
Most Linux distributions include a bundled syslogd(8) daemon that can act as a syslog receiver, but the remote receive capability is disabled by default. To enable Linux to receive syslog traffic, edit the /etc/sysconfig/syslog configuration file such that the `-r` option is included (enables remote logging):

```
SYSLOGD_OPTIONS="-r -m 0"
```

and then restart the logging service:

```
#/etc/init.d/syslog stop
#/etc/init.d/syslog start
```

Some Linux distributions have an ipfilter packet filter that will reject syslog UDP packets by default, and the filter must be modified to permit them. On these distributions, use a command similar to the following to add an INPUT rule to accept syslog UDP packets:

```
# iptables -I INPUT 1 -p udp --sport 514 --dport 514 -j ACCEPT
```

By default, Linux syslogd records messages to /var/log/messages and a test alert would be recorded as follows:

```
```

**Working with System Identity**

This service provides configuration for the system name and location. You might need to change these if the appliance is moved to a different network location, or repurposed.

**TABLE 106**  System Identity Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Name</td>
<td>A single canonical identifying name for the appliance that is shown in the user interface. This name is separate from any DNS names that are used to connect to the system (which would be configured on remote DNS servers). This name can be changed at any time.</td>
</tr>
<tr>
<td>System Location</td>
<td>A text string to describe where the appliance is physically located. If SNMP is enabled, this will be exported as the <code>syslocation</code> string in MIB-II.</td>
</tr>
</tbody>
</table>
Changing services properties is documented in “Services BUI Page Icons and Buttons” on page 177 and “Configuring Services using the CLI” on page 178. The CLI property names are shorter versions of those listed above.

### TABLE 107 System Identity Logs

<table>
<thead>
<tr>
<th>Log</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>system-identity:node</td>
<td>Logs the System Identity service events and errors</td>
</tr>
</tbody>
</table>

## Working with SSH

The SSH (Secure Shell) service allows users to login to the appliance CLI and perform most of the same administrative actions that can be performed in the BUI. The SSH service can also be used as means of executing automated scripts from a remote host, such as for retrieving daily logs or Analytics statistics.

To work with SSH, see the following sections:

- “SSH Properties” on page 259
- “SSH Logs” on page 259
- “Disabling root SSH Access using the CLI” on page 260

### SSH Properties

### TABLE 108 SSH Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server key length</td>
<td>The number of bits in the ephemeral key.</td>
<td>768</td>
</tr>
<tr>
<td>Key regeneration interval</td>
<td>Ephemeral key regeneration interval, in seconds.</td>
<td>3600</td>
</tr>
<tr>
<td>Login grace period</td>
<td>The SSH connection will be disconnected after this many seconds if the client has failed to authenticate.</td>
<td>120</td>
</tr>
<tr>
<td>Permit root login</td>
<td>Allows the root user to login using SSH.</td>
<td>yes</td>
</tr>
</tbody>
</table>

### SSH Logs

### TABLE 109 SSH Logs

<table>
<thead>
<tr>
<th>Log</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network-ssh:default</td>
<td>Log of the SSH service events and errors</td>
</tr>
</tbody>
</table>
Disabling root SSH Access using the CLI

1. Set permit root login to false.
2. Apply/commit the configuration.
Maintaining the Oracle ZFS Storage Appliance

The Oracle ZFS Storage Appliance family of products provides efficient file and block data services to clients over a network, and a rich set of data services that can be applied to the data stored on the system.

Working with Maintenance Workflows

A workflow is a CLI script that is uploaded to and managed by the appliance by itself. Workflows can be parameterized and executed in a first-class fashion from either the browser interface or the command line interface. Workflows may also be optionally executed as alert or at a designated time. As such, workflows allow for the appliance to be extended in ways that capture specific policies and procedures, and can be used (for example) to formally encode best practices for a particular organization or application.

To work with workflows, use the following sections:

- “Understanding Workflows” on page 262
- “Understanding Workflow Parameters” on page 263
- “Constrained Workflow Parameters” on page 264
- “Optional Workflow Parameters” on page 265
- “Workflow Error Handling” on page 266
- “Workflow Input Validation” on page 267
- “Workflow Execution Auditing and Reporting” on page 268
- “Understanding Workflow Versioning” on page 270
- “Using Workflows for Alert Actions” on page 271
- “Using Scheduled Workflows” on page 273
- “Using a Scheduled Workflow” on page 273
- “Coding Workflow Schedules” on page 274
- “Creating a Worksheet Based on a Specified Drive Type” on page 276
- “Uploading Workflows using the BUI” on page 278
- “Downloading Workflows using the CLI” on page 279
- “Listing Workflows using the CLI” on page 279
- “Executing Workflows using the CLI” on page 280
Understanding Workflows

A workflow is embodied in a valid ECMAScript file, containing a single global variable, workflow. This is an Object that must contain at least three members:

**TABLE 110** Required Object Members

<table>
<thead>
<tr>
<th>Required member</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Name of the workflow</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>Description of workflow</td>
</tr>
<tr>
<td>execute</td>
<td>Function</td>
<td>Function that executes the workflow</td>
</tr>
</tbody>
</table>

**EXAMPLE 10** Hello World Workflow

The following is an example of a basic workflow:

```javascript
var workflow = {
    name: 'Hello world',
    description: 'Bids a greeting to the world',
    execute: function () { return ('hello world!') }
};
```

Uploading this workflow will result in a new workflow named "Hello world"; executing the workflow will result in the output "hello world!"

**EXAMPLE 11** Using the Workflow Run Function to Return CPU Utilization

Workflows execute asynchronously in the appliance shell, running (by default) as the user executing the workflow. As such, workflows have at their disposal the appliance scripting facility (see “Working with CLI Scripting” on page 40), and may interact with the appliance just as any other instance of the appliance shell. That is, workflows may execute commands, parse output, modify state, and so on. Here is a more complicated example that uses the run function to return the current CPU utilization:

```javascript
var workflow = {
    name: 'CPU utilization',
    description: 'Displays the current CPU utilization',
    execute: function () {
        run('analytics datasets select name=cpu.utilization');
        cpu = run('csv 1').split('\n')[1].split(',');
        return ('At ' + cpu[0] + ', utilization is ' + cpu[1] + '%');
    }
};
```
Understanding Workflow Parameters

Workflows that do not operate on input have limited scope; many workflows need to be parameterized to be useful. This is done by adding a parameters member to the global workflow object. The parameters member is in turn an object that is expected to have a member for each parameter. Each parameters member must have the following members:

<table>
<thead>
<tr>
<th>Required Member</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>String</td>
<td>Label to adorn input of workflow parameter</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>Type of workflow parameter</td>
</tr>
</tbody>
</table>

The type member must be set to one of these types:

<table>
<thead>
<tr>
<th>Type name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>A boolean value</td>
</tr>
<tr>
<td>ChooseOne</td>
<td>One of a number of specified values</td>
</tr>
<tr>
<td>EmailAddress</td>
<td>An e-mail address</td>
</tr>
<tr>
<td>File</td>
<td>A file to be transferred to the appliance</td>
</tr>
<tr>
<td>Host</td>
<td>A valid host, as either a name or dotted decimal</td>
</tr>
<tr>
<td>HostName</td>
<td>A valid hostname</td>
</tr>
<tr>
<td>HostPort</td>
<td>A valid, available port</td>
</tr>
<tr>
<td>Integer</td>
<td>An integer</td>
</tr>
<tr>
<td>NetAddress</td>
<td>A network address</td>
</tr>
<tr>
<td>NodeName</td>
<td>A name of a network node</td>
</tr>
<tr>
<td>NonNegativeInteger</td>
<td>An integer that is greater than or equal to zero</td>
</tr>
<tr>
<td>Number</td>
<td>Any number -- including floating point</td>
</tr>
<tr>
<td>Password</td>
<td>A password</td>
</tr>
<tr>
<td>Permissions</td>
<td>POSIX permissions</td>
</tr>
<tr>
<td>Port</td>
<td>A port number</td>
</tr>
<tr>
<td>Size</td>
<td>A size</td>
</tr>
<tr>
<td>String</td>
<td>A string</td>
</tr>
<tr>
<td>StringList</td>
<td>A list of strings</td>
</tr>
</tbody>
</table>
EXAMPLE 12  Workflow Using Two Parameters

Based on the specified types, an appropriate input form will be generated upon execution of the workflow. For example, here is a workflow that has two parameters, the name of a business unit (to be used as a project) and the name of a share (to be used as the share name):

```javascript
var workflow = {
    name: 'New share',
    description: 'Creates a new share in a business unit',
    parameters: {
        name: {
            label: 'Name of new share',
            type: 'String'
        },
        unit: {
            label: 'Business unit',
            type: 'String'
        }
    },
    execute: function (params) {
        run('shares select ' + params.unit);
        run('filesystem ' + params.name);
        run('commit');
        return ('Created new share "' + params.name + '"');
    }
};
```

If you upload this workflow and execute it, you will be prompted with a dialog box to fill in the name of the share and the business unit. When the share has been created, a message will be generated indicating as much.

Constrained Workflow Parameters

For some parameters, one does not wish to allow an arbitrary string, but wishes to rather limit input to one of a small number of alternatives. These parameters should be specified to be of type `ChooseOne`, and the object containing the parameter must have two additional members:

| TABLE 113  Constrained Parameters Required Members |
|-----------|------------------|---------------------------------|
| Required Member | Type | Description |
| options | Array | An array of strings that specifies the valid options |
| optionlabels | Array | An array of strings that specifies the labels associated with the options specified in options |
EXAMPLE 13 Using the Workflow ChooseOne Parameter

Using the ChooseOne parameter type, we can enhance the previous example to limit the business unit to be one of a small number of predefined values:

```javascript
var workflow = {
    name: 'Create share',
    description: 'Creates a new share in a business unit',
    parameters: {
        name: {
            label: 'Name of new share',
            type: 'String'
        },
        unit: {
            label: 'Business unit',
            type: 'ChooseOne',
            options: [ 'development', 'finance', 'qa', 'sales' ],
            optionlabels: [ 'Development', 'Finance', 'Quality Assurance', 'Sales/Administrative' ],
        }
    },
    execute: function (params) {
        run('shares select ' + params.unit);
        run('filesystem ' + params.name);
        run('commit');
        return ('Created new share "' + params.name + '"');
    }
};
```

When this workflow is executed, the `unit` parameter will not be entered by hand — it will be selected from the specified list of possible options.

Optional Workflow Parameters

Some parameters may be considered optional in that the UI should not mandate that these parameters are set to any value to allow execution of the workflow. Such a parameter is denoted via the `optional` field of the `parameters` member:

TABLE 114 Required Members for Optional Parameters

<table>
<thead>
<tr>
<th>Optional Member</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>optional</code></td>
<td>Boolean</td>
<td>If set to <code>true</code>, denotes that the parameter need not be set; the UI may allow the workflow to be executed without a value being specified for the parameter.</td>
</tr>
</tbody>
</table>

If a parameter is optional and is unset, its member in the parameters object passed to the `execute` function will be set to `undefined`. 
Workflow Error Handling

If, in the course of executing a workflow, an error is encountered, an exception will be thrown. If the exception is not caught by the workflow itself (or if the workflow throws an exception that is not otherwise caught), the workflow will fail, and the information regarding the exception will be displayed to the user. To properly handle errors, exceptions should be caught and processed. For example, in the previous example, an attempt to create a share in a non-existent project results in an uncaught exception.

EXAMPLE 14      Workflow Error Handling

This example could be modified to catch the offending error, and create the project in the case that it doesn't exist:

```javascript
var workflow = {
    name: 'Create share',
    description: 'Creates a new share in a business unit',
    parameters: {
        name: {
            label: 'Name of new share',
            type: 'String'
        },
        unit: {
            label: 'Business unit',
            type: 'ChooseOne',
            options: [ 'development', 'finance', 'qa', 'sales' ],
            optionlabels: [ 'Development', 'Finance', 'Quality Assurance', 'Sales/Administrative' ],
        }
    },
    execute: function (params) {
        try {
            run('shares select ' + params.unit);
        } catch (err) {
            if (err.code != EAKSH_ENTITY_BADSELECT)
                throw (err);
            /*
            * We haven't yet created a project that corresponds to
            * this business unit; create it now.
            */
            run('shares project ' + params.unit);
            run('commit');
            run('shares select ' + params.unit);
        }
        run('filesystem ' + params.name);
        run('commit');
        return ('Created new share "' + params.name + '"');
    }
};
```
Workflow Input Validation

Workflows may optionally validate their input by adding a validate member that takes as a parameter an object that contains the workflow parameters as members. The validate function should return an object where each member is named with the parameter that failed validation, and each member's value is the validation failure message to be displayed to the user.

EXAMPLE 15  Workflow Input Validation

To extend our example to give a crisp error if the user attempts to create an extant share:

```javascript
var workflow = {
  name: 'Create share',
  description: 'Creates a new share in a business unit',
  parameters: {
    name: {,
      label: 'Name of new share',
      type: 'String'
    },
    unit: {,
      label: 'Business unit',
      type: 'ChooseOne',
      options: [ 'development', 'finance', 'qa', 'sales' ],
      optionlabels: [ 'Development', 'Finance', 'Quality Assurance', 'Sales/Administrative' ]
    }
  },
  validate: function (params) {
    try {
      run('shares select ' + params.unit);
      run('select ' + params.name);
    } catch (err) {
      if (err.code == EAKSH_ENTITY_BADSELECT)
        return;
    }
    return ({ name: 'share already exists' });
  },
  execute: function (params) {
    try {
      run('shares select ' + params.unit);
    } catch (err) {
      if (err.code != EAKSH_ENTITY_BADSELECT)
        throw (err);
      /*
      * We haven't yet created a project that corresponds to
      * this business unit; create it now.
      */
      run('shares project ' + params.unit);
      set('mountpoint', '/export/' + params.unit);
    }
  }
};
```
run('commit');
run('shares select ' + params.unit);
}

run('filesystem ' + params.name);
run('commit');
return ('Created new share "' + params.name + '"');
);

---

**Workflow Execution Auditing and Reporting**

Workflows may emit audit records by calling the `audit` function. The `audit` function's only argument is a string that is to be placed into the audit log.

For complicated workflows that may require some time to execute, it can be useful to provide clear progress to the user executing the workflow. To allow the execution of a workflow to be reported in this way, the `execute` member should return an array of steps. Each array element must contain the following members:

<table>
<thead>
<tr>
<th>Required Member</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>step</td>
<td>String</td>
<td>String that denotes the name of the execution step</td>
</tr>
<tr>
<td>execute</td>
<td>Function</td>
<td>Function that executes the step of the workflow</td>
</tr>
</tbody>
</table>

As with the `execute` function on the workflow as a whole, the `execute` member of each step takes as its argument an object that contains the parameters to the workflow.

**EXAMPLE 16  Workflow Execution Reporting**

As an example, here is a workflow that creates a new project, share, and audit record over three steps:

```javascript
var steps = [
    {
        step: 'Checking for associated project',
        execute: function (params) {
            try {
                run('shares select ' + params.unit);
                catch (err) {
                    if (err.code != EAKSH_ENTITY_BADSELECT) {
                        throw (err);
                    }
                /*
                * We haven't yet created a project that corresponds to
                * this business unit; create it now.
                */
            }
        }
    }
];
```
*/
run('shares project ' + params.unit);
set('mountpoint', '/export/' + params.unit);
run('commit');
run('shares select ' + params.unit);
}
}
}
step: 'Creating share',
execute: function (params) {
run('filesystem ' + params.name);
run('commit');
}
},
step: 'Creating audit record',
execute: function (params) {
audit('created "' + params.name + '" in "' + params.unit);
}
}];

var workflow = {
name: 'Create share',
description: 'Creates a new share in a business unit',
parameters: {
name: {
label: 'Name of new share',
type: 'String'
},
unit: {
label: 'Business unit',
type: 'ChooseOne',
options: ['development', 'finance', 'qa', 'sales'],
optionlabels: ['Development', 'Finance', 'Quality Assurance', 'Sales/Administrative'],
}
},
validate: function (params) {
try {
run('shares select ' + params.unit);
run('select ' + params.name);
} catch (err) {
if (err.code == EAKSH_ENTITY_BADSELECT)
  return;
}

return ({ name: 'share already exists' });
},
execute: function (params) { return (steps); }
};
Understanding Workflow Versioning

There are two aspects of versioning with respect to workflows: the first is the expression of the version of the appliance software that the workflow depends on, and the second is the expression of the version of the workflow itself. Versioning is expressed through two optional members to the workflow:

<table>
<thead>
<tr>
<th>Optional Member</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>required</td>
<td>String</td>
<td>The minimum version of the appliance software required to run this workflow, including the minimum year, month, day, build and branch.</td>
</tr>
<tr>
<td>version</td>
<td>String</td>
<td>Version of this workflow, in dotted decimal (major.minor.micro) form.</td>
</tr>
</tbody>
</table>

**Appliance Versioning** - To express a minimally required version of the appliance software, add the optional `required` field to your workflow. The appliance is versioned in terms of the year, month and day on which the software was built, followed by a build number and then a branch number, expressed as "year.month.day.build-branch". For example "2009.04.10,12-0" would be the twelfth build of the software originally built on April 10th, 2009. To get the version of the current appliance kit software, run the "configuration version get version" CLI command, or look at the "Version" field in the "System" screen in the BUI. Here's an example of using the `required` field:

**EXAMPLE 17** Using the Workflow Required Field

Here's an example of using the `required` field:

```javascript
var workflow = {
  name: 'Configure FC',
  description: 'Configures fibre channel target groups',
  required: '2009.12.25,1-0',
  ...
```

If a workflow requires a version of software that is newer than the version loaded on the appliance, the attempt to upload the workflow will fail with a message explaining the mismatch.

**Workflow Versioning** - In addition to specifying the required version of the appliance software, workflows themselves may be versioned with the `version` field. This string denotes the major, minor and micro numbers of the workflow version, and allows multiple versions of the same workflow to exist on the machine. When uploading a workflow, any `compatible, older` versions of the same workflow are deleted. A workflow is deemed to be `compatible` if it has the
same major number, and a workflow is considered to be older if it has a lower version number. Therefore, uploading a workflow with a version of "2.1" will remove the same workflow with version "2.0" (or version "2.0.1") but not "1.2" or "0.1".

**Using Workflows for Alert Actions**

Workflows may be optionally executed as an alert. To allow a workflow to be eligible as an alert action, its alert action must be set to true.

When executed as alert actions, workflows assume the identity of the user that created them. For this reason, any workflow that is to be eligible as an alert action must set setid to true. Alert actions have a single object parameter that has the following members:

<table>
<thead>
<tr>
<th>Required Member</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>String</td>
<td>The class of the alert.</td>
</tr>
<tr>
<td>code</td>
<td>String</td>
<td>The code of the alert.</td>
</tr>
<tr>
<td>items</td>
<td>Object</td>
<td>An object describing the alert.</td>
</tr>
<tr>
<td>timestamp</td>
<td>Date</td>
<td>Time of alert.</td>
</tr>
</tbody>
</table>

The items member of the parameters object has the following members:

<table>
<thead>
<tr>
<th>Required Member</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>String</td>
<td>The URL of the web page describing the alert action</td>
</tr>
<tr>
<td>action</td>
<td>String</td>
<td>The action that should be taken by the user in response to the alert.</td>
</tr>
<tr>
<td>impact</td>
<td>String</td>
<td>The impact of the event that precipitated the alert.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable string describing the alert.</td>
</tr>
<tr>
<td>severity</td>
<td>String</td>
<td>The severity of the event that precipitated the alert.</td>
</tr>
</tbody>
</table>

Workflows executing as alert actions may use the audit function to generate audit log entries. It is recommended that any relevant debugging information be generated to the audit log via the audit function. For example, here is a workflow that executes failover if in the clustered state -- but audits any failure to reboot:
EXAMPLE 18  Workflow Auditing Failure to Reboot

For example, here is a workflow that executes failover if in the clustered state -- but audits any failure to reboot:

```javascript
var workflow = {
    name: 'Failover',
    description: 'Fail the node over to its clustered peer',
    alert: true,
    setid: true,
    execute: function (params) {
        /*
        * To failover, we first confirm that clustering is configured
        * and that we are in the clustered state. We then reboot,
        * which will force our peer to takeover. Note that we're
        * being very conservative by only rebooting if in the
        * AKCS_CLUSTERED state: there are other states in which it
        * may well be valid to failback (e.g., we are in AKCS_OWNER,
        * and our peer is AKCS_STRIPPED), but those states may also
        * indicate aberrant operation, and we therefore refuse to
        * failback. (Even in an active/passive clustered config, a
        * FAILBACK should always be performed to transition the
        * cluster peers from OWNER/STRIPPED to CLUSTERED/CLUSTERED.)
        */
        var uuid = params.uuid;
        var clustered = 'AKCS_CLUSTERED';

        audit('attempting failover in response to alert ' + uuid);

        try {
            run('configuration cluster');
        } catch (err) {
            audit('could not get clustered state; aborting');
            return;
        }

        if ((state = get('state')) != clustered) {
            audit('state is ' + state + '; aborting');
            return;
        }

        if ((state = get('peer_state')) != clustered) {
            audit('peer state is ' + state + '; aborting');
            return;
        }

        run('cd /');
        run('confirm maintenance system reboot');
    }
};
```
Using Scheduled Workflows

Workflows can be started via a timer event by setting up a schedule for them. The property scheduled has to be added to the Workflow Object and needs to be set to true. Schedules can either be created via the CLI once a workflow is loaded into the appliance or an array type property named schedule can be added to the Object Workflow.

Each schedule entry consists of the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>String</td>
<td>Name of the schedule, system generated</td>
</tr>
<tr>
<td>frequency</td>
<td>String</td>
<td>minute,halfhour,hour,day,week, month</td>
</tr>
<tr>
<td>day</td>
<td>String</td>
<td>Specifies specific day and can be set to: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday or Sunday. Can be set when frequency is set to week or month</td>
</tr>
<tr>
<td>hour</td>
<td>String</td>
<td>00-23, Specifies the hour part of the schedule and can be specified when the frequency is set to a day, week or month.</td>
</tr>
<tr>
<td>minute</td>
<td>String</td>
<td>00-59, Specifies the minute part of the schedule.</td>
</tr>
</tbody>
</table>

Using a Scheduled Workflow

Once a workflow has been loaded into the appliance a schedule can be defined for it via the CLI interface as follows:

**EXAMPLE 19**  Scheduled Workflow in the CLI

dory:> maintenance workflows
  dory:maintenance workflows> "select workflow-002'''
  dory:maintenance workflow-002> schedules
  dory:maintenance workflow-002 schedules>create
  dory:maintenance workflow-002 schedule (uncommitted)> set frequency=day
    frequency = day (uncommitted)
  dory:maintenance workflow-002 schedule (uncommitted)> set hour=10
    hour = 10 (uncommitted)
  dory:maintenance workflow-002 schedule (uncommitted)> set minute=05
    minute = 05 (uncommitted)
  dory:maintenance workflow-002 schedule (uncommitted)> commit
  dory:maintenance workflow-002 schedules> list
Working with Maintenance Workflows

---

NAME FREQUENCY DAY HH:MM
schedule-001 day - 10:05
dory:maintenance workflow-002 schedules> create
dory:maintenance workflow-002 schedule (uncommitted)> set frequency=week
  frequency = week (uncommitted)
dory:maintenance workflow-002 schedule (uncommitted)> set day=Monday
  day = Monday (uncommitted)
dory:maintenance workflow-002 schedule (uncommitted)> set hour=13
  hour = 13 (uncommitted)
dory:maintenance workflow-002 schedule (uncommitted)> set minute=15
  minute = 15 (uncommitted)
dory:maintenance workflow-002 schedule (uncommitted)> commit
dory:maintenance workflow-002 schedules>

NAME FREQUENCY DAY HH:MM
schedule-001 day - 10:05
schedule-002 week Monday 13:15
dory:maintenance workflow-002 schedules>

Coding Workflow Schedules

Schedules can also be specified in the workflow code as a property in the Object workflow. The property syntax used here differs from the CLI schedule creation. Here three properties are used,

TABLE 120  Workflow Schedule Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>offset</td>
<td>Number</td>
<td>Determines the starting point in the defined period</td>
</tr>
<tr>
<td>period</td>
<td>Number</td>
<td>Defines the frequency of the Schedule</td>
</tr>
<tr>
<td>unit</td>
<td>String</td>
<td>Specifies if either seconds or month are used as unit in the offset and period definition</td>
</tr>
</tbody>
</table>

EXAMPLE 20  Illustrating the Use of Workflow Properties

The following code example illustrates the use of the properties. Note that inline arithmetic helps to make the offset and period declarations more readable.

```javascript
// Example of using Schedule definitions within a workflow
var MyTextObject = {
  MyVersion: '1.0',
  MyName: 'Example 9',
  MyDescription: 'Example of use of Timer',
  Origin: 'Oracle'
};
var MySchedules = [
  // half hr interval
```
Working with Maintenance Workflows

```javascript
var workflow = {
  name: 'MyTextObject.MyName',
  description: 'MyTextObject.MyDescription',
  version: 'MyTextObject.MyVersion',
  alert: false,
  setid: true,
  schedules: MySchedules,
  scheduled: true,
  origin: 'MyTextObject.Origin',
  execute: function () {
    audit('workflow started for timer; ');
  }
};
```

The property `units` in the Object `MySchedules` specifies the type of units used for the properties `offset` and `period`. They can be set to either `seconds` or `month`. The property `period` specifies the frequency of the event and the offset specifies the units within the period. In the above example the period in the second schedule is set for a week, starting at the second day, at 4:30. Multiple schedules can be defined in the property `schedules`.

The Object `MySchedules` in the example uses the following three properties:

- **offset** - This is the starting offset from January 1, 1970 for the schedule. The offset is given in the units defined by the property "units".
- **period** - This is the period between recurrences of the schedule which is also given in the units defined by the property "units."
- **units** - This can be defined in seconds or months.

The starting point for weekly schedules is Thursday. This is due to the fact that the epoch is defined as starting on 1 Jan 1970 which was a Thursday.

**EXAMPLE 21** Workflow Schedule Shown in the CLI

In the above example the period in the second schedule uses a starting offset of 2 days + 4 hours + 30 minutes. This results in the starting date being January 3, 1970 at 4:30 am. The schedule recurs weekly indefinitely every Saturday at 4:30 am. Below you can see the display of the schedule in the CLI.

```
<small>dory:> maintenance workflows
dory:maintenance workflows> list
WORKFLOW     NAME                       OWNER SETID ORIGIN               VERSION
workflow-000 Configure for Oracle Solaris Cluster NFS root  false Oracle Corporation   1.0.0
workflow-001 Unconfigure Oracle Solaris Cluster NFS root  false Oracle Corporation   1.0.0
workflow-002 Configure for Oracle Enterprise Manager Monitoring root  false Sun Microsystems, Inc. 1.1
workflow-003 Unconfigure Oracle Enterprise Manager Monitoring root  false Sun Microsystems, Inc. 1.0</small>
```
Creating a Worksheet Based on a Specified Drive Type

Here is an example workflow that creates a worksheet based on a specified drive type:

**EXAMPLE 22**  Workflow Device Type Selection

```javascript
var steps = [
    {step: 'Checking for existing worksheet',
      execute: function (params) {
        /*
        * In this step, we're going to see if the worksheet that
        * we're going to create already exists. If the worksheet
        * already exists, we blow it away if the user has indicated
        * that they desire this behavior. Note that we store our
        * derived worksheet name with the parameters, even though
        * it is not a parameter per se; this is explicitly allowed,
        * and it allows us to build state in one step that is
        * processed in another without requiring additional global
        * variables.
        */
        params.worksheet = 'Drilling down on ' + params.type + ' disks';
        try {
            run('analytics worksheets select name="' +
                params.worksheet + '"');
            if (params.overwrite) {
                run('confirm destroy');
                return;
            }
            throw ('Worksheet called "' +
                params.worksheet + '" already exists!');
        } catch (err) {
            if (err.code != EAKSH_ENTITY_BADSELECT)
                throw (err);
        }
    },
    {step: 'Finding disks of specified type',
      execute: function (params) {
        /*
        * In this step, we will iterate over all chassis, and for
        */
    }
];
```
* each chassis iterates over all disks in the chassis, looking for disks that match the specified type.
*/
var chassis, name, disks;
var i, j;

run('cd /');
run('maintenance hardware');

chassis = list();
params.disks = [];

for (i = 0; i < chassis.length; i++) {
  run('select ' + chassis[i]);
  name = get('name');
  run('select disk');
  disks = list();

  for (j = 0; j < disks.length; j++) {
    run('select ' + disks[j]);
    if (get('use') == params.type) {
      params.disks.push(name + '/' + get('label'));
    }
    run('cd ..');
  }
  run('cd ../..');
}

if (params.disks.length === 0)
  throw ('No ' + params.type + ' disks found');
run('cd /');
}

}, {
  step: 'Creating worksheet',
  execute: function (params) {
    /*
    * In this step, we're ready to actually create the worksheet itself: we have the disks of the specified type and
    * we know that we can create the worksheet. Note that we create several datasets: first, I/O bytes broken down
    * by disk, with each disk of the specified type highlighted as a drilldown. Then, we create a separate dataset for
    * each disk of the specified type. Finally, note that we aren't saving the datasets -- we'll let the user do that
    * from the created worksheet if they so desire. (It would be straightforward to add a boolean parameter to this
    * workflow that allows that last behavior to be optionally changed.)
    */
var disks = [], i;

run('analytics worksheets');
run('create "' + params.worksheet + '"');
run('select name="' + params.worksheet + '"');
run('dataset');
run('set name=io.bytes[disk]');

for (i = 0; i < params.disks.length; i++)
  disks.push('"' + params.disks[i] + '"');

run('set drilldown=' + disks.join(','));
run('commit');

for (i = 0; i < params.disks.length; i++) {
  run('dataset');
  run('set name="io.bytes[disk=' + params.disks[i] + ']"');
  run('commit');
}

var workflow = {
  name: 'Disk drilldown',
  description: 'Creates a worksheet that drills down on system, ' +
                'cache, or log devices',
  parameters: {
    type: {
      label: 'Create a new worksheet drilling down on',
      type: 'ChooseOne',
      options: [ 'cache', 'log', 'system' ],
      optionlabels: [ 'Cache', 'Log', 'System' ]
    },
    overwrite: {
      label: 'Overwrite the worksheet if it exists',
      type: 'Boolean'
    },
  },
  execute: function (params) { return (steps); }
};

Uploading Workflows using the BUI

Workflows are uploaded to the appliance by clicking on the plus icon, and they are executed by clicking on the row specifying the workflow.
Downloading Workflows using the CLI

1. Workflows are downloaded to the appliance via the `download` command, which is similar to the mechanism used for software updates:

   dory:maintenance workflows> download
   dory:maintenance workflows download (uncommitted)> get
     url = (unset)
     user = (unset)
     password = (unset)

2. You must set the "url" property to be a valid URL for the workflow. This may be either local to your network or over the internet. The URL can be either HTTP (beginning with "http://") or FTP (beginning with "ftp://"). If user authentication is required, it may be a part of the URL (e.g. "ftp://myusername:mypasswd@myserver/export/foo"), or you may leave the username and password out of the URL and instead set the user and password properties.

   dory:maintenance workflows download (uncommitted)> set url=ftp://foo/example1.akwf
   url = ftp://foo/example1.akwf
   dory:maintenance workflows download (uncommitted)> set user=bmc
     user = bmc
   dory:maintenance workflows download (uncommitted)> set password
     Enter password:
     password = ********
   dory:maintenance workflows download (uncommitted)> commit
   Transferred 138 of 138 (100%) ... done

Listing Workflows using the CLI

1. To list workflows, use the `list` command from the `maintenance workflows` context:
Executing Workflows using the CLI

1. To execute a workflow, use the `execute` command from within the context of the selected workflow. If the workflow takes no parameters, it will simply execute:

```
dory:maintenance workflow-000> execute
hello world!
```
2. If the workflow takes parameters, the context will become a captive context in which parameters must be specified:

```
  dory:maintenance workflow-000> execute
  dory:maintenance workflow-000 execute (uncommitted)> get
    type = (unset)
    overwrite = (unset)
```

3. Any attempt to commit the execution of the workflow without first setting the requisite parameters will result in an explicit failure:

```
  dory:maintenance workflow-000 execute (uncommitted)> commit
  error: cannot execute workflow without setting property "type"
```

4. To execute the workflow, set the specified parameters, and then use the commit command:

```
  dory:maintenance workflow-000 execute (uncommitted)> set type=system
  type = system
  dory:maintenance workflow-000 execute (uncommitted)> set overwrite=true
  overwrite = true
  dory:maintenance workflow-000 execute (uncommitted)> commit
```

5. If the workflow has specified steps, those steps will be displayed via the CLI, e.g.:

```
  dory:maintenance workflow-000 execute (uncommitted)> commit
  Checking for existing worksheet ... done
  Finding disks of specified type ... done
  Creating worksheet ... done
```
Working with Shares

This section describes Oracle ZFS Storage Appliance shares, projects and schema. For common administrative purposes, including space management and common settings, shares can be grouped into appliance projects. In addition to the standard built in properties, you can configure any number of additional properties that are available on all shares and projects. These properties are given basic types for validation purposes, and are inherited like most other standard properties. The values are never consumed by the software in any way, and exist solely for end-user consumption. The property schema is global to the system, across all pools, and is synchronized between cluster peers.

Shares are filesystems and LUNs that are exported over supported data protocols to clients of the appliance. Filesystems export a file-based hierarchy and can be accessed over SMB, NFS, HTTP/WebDav, and FTP. LUNs export block-based volumes and can be accessed over iSCSI or Fibre Channel. The project/share tuple is a unique identifier for a share within a pool. Multiple projects can contain shares with the same name, but a single project cannot contain shares with the same name. A single project can contain both filesystems and LUNs, and they share the same namespace.

To configure shares, projects, or schema, use the following sections:

- “Working with Storage Pools” on page 284
- “Share Properties” on page 285
- “Working with Snapshots” on page 286
- “Working with Clones” on page 286
- “Space Management for Shares” on page 287
- “Working with Filesystem Namespace” on page 294
- “Working with the Shares > Shares BUI Page” on page 295
- “Working with the Shares > Shares > General BUI Page” on page 309
- “Understanding the Shares > Shares > Protocols BUI Page” on page 314
- “Shares Protocols - SFTP” on page 321
- “Working with the Shares > Shares > Access BUI Page” on page 321
- “Shares - Snapshots” on page 328
- “Working with Replication” on page 337
- “Working with Shadow Migration” on page 374
- “Working with Projects” on page 382
- “Working with Schemas” on page 391
- “Working with Data Encryption” on page 394
Working with Storage Pools

The appliance is based on the ZFS filesystem. ZFS groups underlying storage devices into pools, and filesystems and LUNs allocate from this storage as needed. Before creating filesystems or LUNs, you must first configure storage on the appliance. Once a storage pool is configured, there is no need to statically size filesystems, though this behavior can be achieved by using “Space Management for Shares” on page 287.

While multiple storage pools are supported, this type of configuration is generally discouraged because it provides significant drawbacks as described in “Storage Configuration” on page 77. Multiple pools should only be used where the performance or reliability characteristics of two different profiles are drastically different, such as a mirrored pool for databases and a RAID-Z pool for streaming workloads.

When multiple pools are active on a single host, the BUI will display a drop-down list in the menu bar that can be used to switch between pools. In the CLI, the name of the current pool
will be displayed in parenthesis, and can be changed by setting the 'pool' property. If there is only a single pool configured, then these controls will be hidden. When multiple pools are selected, the default pool chosen by the UI is arbitrary, so any scripted operation should be sure to set the pool name explicitly before manipulating any shares.

Share Properties

All projects and shares have a number of associated properties. These properties fall into the following groups:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherited</td>
<td>This is the most common type of property, and represents most of the configurable project and share properties. Shares that are part of a project can either have local settings for properties, or they can inherit their settings from the parent project. By default, shares inherit all properties from the project. If a property is changed on a project, all shares that inherit that property are updated to reflect the new value. When inherited, all properties have the same value as the parent project, with the exception of the mount point and SMB properties. When inherited, these properties concatenate the project setting with their own share name.</td>
</tr>
<tr>
<td>Read-only</td>
<td>These properties represent statistics about the project and share and cannot be changed. The most common properties of this type are space usage statistics.</td>
</tr>
<tr>
<td>Space Management</td>
<td>These properties (quota and reservation) apply to both shares and projects, but are not inherited. A project with a quota of 100G will be enforced across all shares, but each individual share will have no quota unless explicitly set.</td>
</tr>
<tr>
<td>Create time</td>
<td>These properties can be specified at filesystem or LUN creation time, but cannot be changed once the share has been created. These properties control the on-disk data structures, and include internationalization settings, case sensitivity, and volume block size.</td>
</tr>
<tr>
<td>Project default</td>
<td>These properties are set on a project, but do not affect the project itself. They are used to populate the initial settings when creating a filesystem or LUN, and can be useful when shares have a common set of non-inheritable properties. Changing these properties do not affect existing shares, and the properties can be changed before or after creating the share.</td>
</tr>
<tr>
<td>Filesystem local</td>
<td>These properties apply only to filesystems, and are convenience properties for managing the root directory of the filesystem. They cannot be set on projects. These access control properties can also be set by in-band protocol operations.</td>
</tr>
</tbody>
</table>
Working with Snapshots

A snapshot is a point-in-time copy of a filesystem or LUN. Snapshots can be created manually or by setting up an automatic schedule. Snapshots initially consume no additional space, but as the active share changes, previously unreferenced blocks will be kept as part of the last snapshot. Over time, the last snapshot will take up additional space, with a maximum equivalent to the size of the filesystem at the time the snapshot was taken.

Filesystem snapshots can be accessed over the standard protocols in the .zfs/snapshot directory at the root of the filesystem. This directory is hidden by default, and can only be accessed by explicitly changing to the .zfs directory. This behavior can be changed in the snapshot view, but may cause backup software to backup snapshots in addition to live data. LUN Snapshots cannot be accessed directly, though they can be used as a rollback target or as the source of a clone. Project snapshots are the equivalent of snapshotting all shares within the project, and snapshots are identified by name. If a share snapshot that is part of a larger project snapshot is renamed, it will no longer be considered part of the same snapshot, and if any snapshot is renamed to have the same name as a snapshot in the parent project, it will be treated as part of the project snapshot.

Shares support the ability to rollback to previous snapshots. When a rollback occurs, any newer snapshots (and clones of newer snapshots) will be destroyed, and the active data will be reverted to the state when the snapshot was taken. Snapshots only include data, not properties, so any property settings changed since the snapshot was taken will remain.

Working with Clones

A clone is a writable copy of a share snapshot, and is treated as an independent share for administrative purposes. Like snapshots, a clone will initially take up no extra space, but as new data is written to the clone, the space required for the new changes will be associated with the
clone. Clones of projects are not supported. Because space is shared between snapshots and clones, and a snapshot can have multiple clones, a snapshot cannot be destroyed without also destroying any active clones.

Space Management for Shares

The behavior of filesystems and LUNs with respect to managing physical storage is different on the Oracle ZFS Storage Appliance than on many other systems. As described in “Working with Shares”, the appliance leverages a pooled storage model where all filesystems and LUNs share common space. Filesystems never have an explicit size assigned to them, and only take up as much space as they need. LUNs reserve enough physical space to write the entire contents of the device, unless they are thinly provisioned, in which case they behave like filesystems and use only the amount of space physically consumed by data.

This system provides maximum flexibility and simplicity of management in an environment when users are generally trusted to do the right thing. A stricter environment, where user's data usage is monitored and/or restricted, requires more careful management. This section describes some of the tools available to the administrator to control and manage space usage.

To work with shares space management, see the following sections:

- “Shares Terminology” on page 287
- “Understanding Snapshots” on page 288
- “File System and Project Settings” on page 289
- “Viewing Current Shares Usage in the BUI” on page 290
- “Viewing Current Shares Usage in the CLI” on page 291
- “Setting User or Group Quotas” on page 291
- “Setting User or Group Quotas using the BUI” on page 292
- “Setting User or Group Quotas using the CLI” on page 293
- “Working with Identity Management” on page 294

Shares Terminology

Before getting into details, it is important to understand some basic terms used when talking about space usage on the appliance.

- **Physical Data** - Size of data as stored physically on disk. Typically, this is equivalent to the logical size of the corresponding data, but can be different in the phase of compression or other factors. This includes the space of the active share as well as all snapshots. Space accounting is generally enforced and managed based on physical space.

- **Logical Data** - The amount of space logically consumed by a filesystem. This does not factor into compression, and can be viewed as the theoretical upper bound on the amount of space consumed by the filesystem. Copying the filesystem to another appliance using
a different compression algorithm will not consume more than this amount. This statistic is not explicitly exported and can generally only be computed by taking the amount of physical space consumed and multiplying by the current compression ratio.

■ **Referenced Data** - This represents the total amount of space referenced by the active share, independent of any snapshots. This is the amount of space that the share would consume should all snapshots be destroyed. This is also the amount of data that is directly manageable by the user over the data protocols.

■ **Snapshot Data** - This represents the total amount of data currently held by all snapshots of the share. This is the amount of space that would be free should all snapshots be destroyed.

■ **Quota** - A quota represents a limit on the amount of space that can be consumed by any particular entity. This can be based on filesystem, project, user, or group, and is independent of any current space usage.

■ **Reservation** - A reservation represents a guarantee of space for a particular project or filesystem. This takes available space away from the rest of the pool without increasing the actual space consumed by the filesystem. This setting cannot be applied to users and groups. The traditional notion of a statically sized filesystem can be created by setting a quota and reservation to the same value.

---

### Understanding Snapshots

Snapshots present an interesting dilemma for space management. They represent the set of physical blocks referenced by a share at a given point in time. Initially, this snapshot consumes no additional space. But as new data is overwritten in the new share, the blocks in the active share will only contain the new data, and older blocks will be “held” by the most recent (and possibly older) snapshots. Gradually, snapshots can consume additional space as the content diverges in the active share.

Some other systems will try to hide the cost of snapshots, by pretending that they are free, or by "reserving" space dedicated to holding snapshot data. Such systems try to gloss over the basic fact inherent with snapshots. If you take a snapshot of a filesystem of any given size, and re-write 100% of the data within the filesystem, by definition you must maintain references to twice the data as was originally in the filesystem. Snapshots are not free, and the only way other systems can present this abstraction is to silently destroy snapshots when space gets full. This can often be the absolute worst thing to do, as a process run amok rewriting data can cause all previous snapshots to be destroyed, preventing any restoration in the process.

In the Oracle ZFS Storage Appliance, the cost of snapshots is always explicit, and tools are provided to manage this space in a way that best matches the administrative model for a given environment. Each snapshot has two associated space statistics: unique space and referenced space. The amount of referenced space is the total space consumed by the filesystem at the time the snapshot was taken. It represents the theoretical maximum size of the snapshot should it remain the sole reference to all data blocks. The unique space indicates the amount of physical space referenced only by the current snapshot. When a snapshot is destroyed, the unique space will be made available to the rest of the pool. Note that the amount of space consumed by all
snapshots is not equivalent to the sum of unique space across all snapshots. With a share and a single snapshot, all blocks must be referenced by one or both of the snapshot or the share. With multiple snapshots, however, it's possible for a block to be referenced by some subset of snapshots, and not any particular snapshot. For example, if a file is created, two snapshots X and Y are taken, the file is deleted, and another snapshot Z is taken, the blocks within the file are held by X and Y, but not by Z. In this case, destroying Z will not free up the space, but destroying both X and Y will. Because of this, destroying any snapshot can affect the unique space referenced by neighboring snapshots, though the total amount of space consumed by snapshots will always decrease.

The total size of a project or share always accounts for space consumed by all snapshots, though the usage breakdown is also available. Quotas and reservations can be set at the project level to enforce physical constraints across this total space. In addition, quotas and reservations can be set at the filesystem level, and these settings can apply to only referenced data or total data. Whether or not quotas and reservations should be applied to referenced data or total physical data depends on the administrative environment. If users are not in control of their snapshots (i.e. an automatic snapshot schedule is set for them), then quotas should typically not include snapshots in the calculation. Otherwise, the user may run out of space but be confused when files cannot be deleted. Without an understanding of snapshots or means to manage those snapshots, it is possible for such a situation to be unrecoverable without administrator intervention. In this scenario, the snapshots represent an overhead cost that is factored into operation of the system in order to provide backup capabilities. On the other hand, there are environments where users are billed according to their physical space requirements, and snapshots represent a choice by the user to provide some level of backup that meets their requirements given the churn rate of their dataset. In these environments, it makes more sense to enforce quotas based on total physical data, including snapshots. The users understand the cost of snapshots, and can be provided a means to actively management them (as through dedicated roles on the appliance).

**File System and Project Settings**

The simplest way of enforcing quotas and reservations is on a per-project or per-filesystem basis. Quotas and reservations do not apply to LUNs, though their usage is accounted for in the total project quota or reservations.

A data quota enforces a limit on the amount of space a filesystem or project can use. By default, it will include the data in the filesystem and all snapshots. Clients attempting to write new data will get an error when the filesystem is full, either because of a quota or because the storage pool is out of space. As described in “Understanding Snapshots” on page 288, this behavior may not be intuitive in all situations, particularly when snapshots are present. Removing a file may cause the filesystem to write new data if the data blocks are referenced by a snapshot, so it may be the case that the only way to decrease space usage is to destroy existing snapshots.

If the 'include snapshots' property is unset, then the quota applies only to the immediate data referenced by the filesystem, not any snapshots. The space used by snapshots is enforced by the
project-level quota but is otherwise not enforced. In this situation, removing a file referenced by a snapshot will cause the filesystem's referenced data to decrease, even though the system as a whole is using more space. If the storage pool is full (as opposed to the filesystem reaching a preset quota), then the only way to free up space may be to destroy snapshots.

Data quotas are strictly enforced, which means that as space usage nears the limit, the amount of data that can be written must be throttled as the precise amount of data to be written is not known until after writes have been acknowledged. This can affect performance when operating at or near the quota. Because of this, it is generally advisable to remain below the quota during normal operating procedures.

Quotas are managed through the BUI under Shares > General > Space Usage > Data. They are managed in the CLI as the quota and quota_snap properties.

A data reservation is used to make sure that a filesystem or project has at least a certain amount of available space, even if other shares in the system try to use more space. This unused reservation is considered part of the filesystem, so if the rest of the pool (or project) reaches capacity, the filesystem can still write new data even though other shares may be out of space.

By default, a reservation includes all snapshots of a filesystem. If the 'include snapshots' property is unset, then the reservation only applies to the immediate data of the filesystem. The behavior when taking snapshots may not always be intuitive. If a reservation on filesystem data (but not snapshots) is in effect, then whenever a snapshot is taken, the system must reserve enough space for that snapshot to diverge completely, even if that never occurs. For example, if a 50G filesystem has a 100G reservation without snapshots, then taking the first snapshot will reserve an additional 50G of space, and the filesystem will end up reserving 150G of space total. If there is insufficient space to guarantee complete divergence of data, then taking the snapshot will fail.

Reservations are managed through the BUI under Shares > General > Space Usage > Data. They are managed in the CLI as the reservation and reservation_snap properties.

When you create a LUN the full physical space you configure for the LUN is reserved and cannot be used by other file systems (unless it is thinly provisioned). For replication, when you take a snapshot of a LUN of any given size, up to twice the size of the LUN is also reserved, depending on how much of the LUN space has been used.

The following list shows the maximum overhead space required when replicating a LUN:

- Up to 100% on the source between updates
- Up to 200% on the source during an update
- Up to 200% on the target

Viewing Current Shares Usage in the BUI

1. To view the current usage in the BUI, go to Shares > Shares > General.
2. **In the Space Usage - Users and Groups section** click the User or Group drop down to select User or Group and query the current usage for any given user or group within a share or across a project.

3. **Type the name of the User or Group that you want to query.** The query progresses as you type.

When the lookup is completed, the current usage is displayed. In addition, the “Show All” link shows a dialog with a list of current usage of all users or groups. This dialog can only query for a particular type - users or groups - and does not support querying both at the same time. This list displays the canonical UNIX and Windows name (if mappings are enabled), as well as the usage and (for filesystems) quota.

### Viewing Current Shares Usage in the CLI

1. **In the CLI, use the `users` and `groups` commands in the context of a particular project or share.**

2. **Use, the `show` command to display current usage in a tabular form.**

3. **To retrieve the usage for a particular user or group, select the user or group you want and use the `get` command.**

```bash
clownfish:> shares select default
clownfish:shares default> users
clownfish:shares default users> list
USER    NAME     USAGE
user-000 root     325K
user-001 ahl     9.94K
user-002 eschrock 20.0G
clownfish:shares default users> select name=eschrock
clownfish:shares default user-002> get
    name = eschrock
    unixname = eschrock
    unixid = 132651
    winname = (unset)
    winid = (unset)
    usage = 20.0G
```

### Setting User or Group Quotas

Quotas can be set on a user or group at the filesystem level, as well as the project level. These enforce physical data usage based on the POSIX or Windows identity of the owner or group of the file or directory. There are some significant differences between user and group quotas and filesystem and project data quotas:
User and group quotas can be applied to filesystems and projects.

Default quotas can be set at the project level and inherited by the project's filesystems.

Default quotas set at the project level can be changed at the filesystem level.

User and group quotas are implemented using *delayed enforcement*. This means that users will be able to exceed their quota for a short period of time before data is written to disk. Once the data has been pushed to disk, the user will receive an error on new writes, just as with the filesystem-level quota case.

User and group quotas are always enforced against referenced data. This means that snapshots do not affect any quotas, and a clone of a snapshot will consume the same amount of effective quota, even though the underlying blocks are shared.

User and group reservations are not supported.

User and group quotas, unlike data quotas, are stored with the regular filesystem data. This means that if the filesystem is out of space, you will not be able to make changes to user and group quotas. You must first make additional space available before modifying user and group quotas.

User and group quotas are sent as part of any remote replication. It is up to the administrator to ensure that the name service environments are identical on the source and destination.

NDMP backup and restore of an entire share will include any user or group quotas. Restores into an existing share will not affect any current quotas.

### Setting User or Group Quotas using the BUI

1. In the BUI, go to Shares > Shares > General.

2. In the Space Usage - Users and Groups section click the User or Group drop down to select User or Group and query the current usage for any given user or group within a share or across a project.

3. In the browser, user quotas are managed from the General tab, under Space Usage -> Users & Groups. As with viewing usage, the current usage is shown as you type a user or group. After you have finished entering the user or group name and the current usage is displayed, the quota can be set, or changed if set at the project level, by selecting one of three options: 1) None - No quota is set for this filesystem; 2) Default - Sets the quota to the default quota at the project level; if no default was set, no quota is set for this filesystem; or 3) Click the radio button, enter a quota in the size field, and select a measurement.

4. After any changes have been set, click Apply.

While all the properties on the page are committed together, the user and group quota are validated separately from the other properties. If an invalid user and group is entered as well as another invalid property, only one of the validation errors may be displayed. After that error has been corrected, an attempt to apply the changes again will show the other error.
Setting User or Group Quotas using the CLI

1. In the CLI, user quotas are managed using the 'users' or 'groups' command from share context. Quotas can be set by selecting a particular user or group and using the 'set quota' command. Any user that is not consuming any space on the filesystem and does not have any quota set will not appear in the list of active users. To set a quota for such a user or group, use the 'quota' command, after which the name and quota can be set. To clear a quota, set it to the value '0'.

2. If a default user or group quota was set at the project level, this procedure overrides that value. When using the list command, the Source column displays "local" if the quota was set at the filesystem level, "default" if set at the project level, or "-" if no quota was set. In the following example, the default user quota set at the project level is 50 GB.

```
clownfish:> shares select default select eschrock
clownfish:shares default/eschrock> users
clownfish:shares default/eschrock users> list
USER     NAME      USAGE  QUOTA  SOURCE
user-000 root      321K      -       -
user-001 ahl       9.94K      -       -
user-002 eschrock  20.0G      -       -
clownfish:shares default/eschrock users> select name=eschrock
clownfish:shares default/eschrock user-002> get
   name = eschrock
   unixname = eschrock
   unixid = 132651
   winname = (unset)
   winid = (unset)
   usage = 20.0G
   quota = (unset)
   source = (unset)
clownfish:shares default/eschrock user-002> set quota=100G
   quota = 100G (uncommitted)
clownfish:shares default/eschrock user-002> commit
clownfish:shares default/eschrock user-002> done
clownfish:shares default/eschrock users> quota (uncommitted)> set name=bmc
   name = bmc (uncommitted)
   quota = default (uncommitted)
clownfish:shares default/eschrock users quota (uncommitted)> set quota=default
   quota = default (uncommitted)
clownfish:shares default/eschrock users> list
USER     NAME      USAGE  QUOTA  SOURCE
user-000 root      321K      -       -
user-001 ahl       9.94K      -       -
user-002 eschrock  20.0G  100G  local
user-003 bmc       -     50G  default
```
Working with Identity Management

User and group quotas leverage the identity mapping service on the appliance. This allows users and groups to be specified as either UNIX or Windows identities, depending on the environment. Like file ownership, these identities are tracked in the following ways:

- If there is no UNIX mapping, a reference to the windows ID is stored.
- If there is a UNIX mapping, then the UNIX ID is stored.

This means that the canonical form of the identity is the UNIX ID. If the mapping is changed later, the new mapping will be enforced based on the new UNIX ID. If a file is created by a Windows user when no mapping exists, and a mapping is later created, new files will be treated as a different owner for the purposes of access control and usage format. This also implies that if a user ID is reused (i.e. a new user name association created), then any existing files or quotas will appear to be owned by the new user name.

It is recommended that any identity mapping rules be established before attempting to actively use filesystems. Otherwise, any change in mapping can sometimes have surprising results.

Working with Filesystem Namespace

Every filesystem on the appliance must be given a unique mountpoint which serves as the access point for the filesystem data. Projects can be given mountpoints, but these serve only as a tool to manage the namespace using inherited properties. Projects are never mounted, and do not export data over any protocol.

All shares must be mounted under /export. While it is possible to create a filesystem mounted at /export, it is not required. If such a share doesn't exist, any directories will be created dynamically as necessary underneath this portion of the hierarchy. Each mountpoint must be unique within a cluster.

- Namespace Nested Mountpoints - It is possible to create filesystems with mountpoints beneath that of other filesystems. In this scenario, the parent filesystems are mounted before children and vice versa. The following cases should be considered when using nested mountpoints:
  - If the mountpoint doesn't exist, one will be created, owned by root and mode 0755. This mountpoint may or may not be torn down when the filesystem is renamed, destroyed, or moved, depending on circumstances. To be safe, mountpoints should be created within the parent share before creating the child filesystem.
  - If the parent directory is read-only, and the mountpoint doesn't exist, the filesystem mount will fail. This can happen synchronously when creating a filesystem, but can also happen asynchronously when making a large-scale change, such as renaming filesystems with inherited mountpoints.
  - When renaming a filesystem or changing its mountpoint, all children beneath the current mountpoint as well as the new mountpoint (if different) will be unmounted.
and remounted after applying the change. This will interrupt any data services currently accessing the share.

- Support for automatically traversing nested mountpoints depends on protocol, as outlined below.

- **Namespace NFSv2 / NFSv3 / NFSv4** - Under NFS, each filesystem is a unique export made visible via the MOUNT protocol. NFSv2 and NFSv3 have no way to traverse nested filesystems, and each filesystem must be accessed by its full path. While nested mountpoints are still functional, attempts to cross a nested mountpoint will result in an empty directory on the client. While this can be mitigated through the use of automount mounts, transparent support of nested mountpoints in a dynamic environment requires NFSv4.

  NFSv4 has several improvements over NFSv3 when dealing with mountpoints. First is that parent directories can be mounted, even if there is no share available at that point in the hierarchy. For example, if \( /\text{export}/\text{home} \) was shared, it is possible to mount \( /\text{export} \) on the client and traverse into the actual exports transparently. More significantly, some NFSv4 clients (including Linux) support automatic client-side mounts, sometimes referred to as "mirror mounts". With such a client, when a user traverses a mountpoint, the child filesystem is automatically mounted at the appropriate local mountpoint, and torn down when the filesystem is unmounted on the client. From the server's perspective, these are separate mount requests, but they are stitched together onto the client to form a seamless filesystem namespace.

- **Namespace SMB** - The SMB protocol does not use mountpoints, as each share is made available by resource name. However, each filesystem must still have a unique mountpoint. Nested mountpoints (multiple filesystems within one resource) are not currently supported, and any attempt to traverse a mountpoint will result in an empty directory.

- **Namespace FTP / FTPS / SFTP** - Filesystems are exported using their standard mountpoint. Nested mountpoints are fully supported and are transparent to the user. However, it is not possible to not share a nested filesystem when its parent is shared. If a parent mountpoint is shared, then all children will be shared as well.

- **Namespace HTTP / HTTPS** - Filesystems are exported under the \( /\text{shares} \) directory, so a filesystem at \( /\text{export}/\text{home} \) will appear at \( /\text{shares}/\text{export}/\text{home} \) over HTTP/HTTPS. Nested mountpoints are fully supported and are transparent to the user. The same behavior regarding conflicting share options described in the FTP protocol section also applies to HTTP.

---

**Working with the Shares > Shares BUI Page**

The Shares UI is accessed from Shares > Shares. The default view shows shares across all projects on the system.

To work with the Shares > Shares BUI Page, see the following sections:

- "Understanding the Shares List" on page 296
Understanding the Shares List

The default view is a list of all shares on the system. This list allows you to rename shares, move shares between projects, and edit individual shares. The shares are divided into two lists, "Filesystems" and "LUNs," that can be selected by switching tabs on this view. The following fields are displayed for each share:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the share. If looking at all projects, this will include the project name as well. The share name is an editable text field. Clicking on the name will allow you to enter a new name. Hitting return or moving focus from the name will commit the change. You will be asked to confirm the action, as renaming shares requires disconnecting active clients.</td>
</tr>
<tr>
<td>Size</td>
<td>For filesystems, this is the total size of the filesystem. For LUNs it is the size of the volume, which may or may not be thinly provisioned.</td>
</tr>
<tr>
<td>Mountpoint</td>
<td>Mountpoint of the filesystem. This is the path available over NFS, and the relative path for FTP and HTTP. Filesystems exported over SMB only use their resource name, though each still need a unique mountpoint somewhere on the system.</td>
</tr>
<tr>
<td>GUID</td>
<td>The SCSI GUID for the LUN. See “Understanding the Shares &gt; Shares &gt; Protocols BUI Page” on page 314 for more information.</td>
</tr>
</tbody>
</table>

The following tools are available for each share:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Move a share to a different project. If the project panel is not expanded, this will automatically expand the panel until the share is dropped onto a project.</td>
</tr>
</tbody>
</table>
### Editing a Share

To edit a share, click on the pencil icon or double-click the row in the share list. This will select the share, and give several different tabs to choose from for editing properties of the share. The complete set of functionality can be found in the section for each tab:

- “Working with the Shares > Shares > General BUI Page” on page 309
- “Understanding the Shares > Shares > Protocols BUI Page” on page 314
- “Working with the Shares > Shares > Access BUI Page” on page 321
- “Shares - Snapshots” on page 328
- “Working with Replication” on page 337
- “Working with Shadow Migration” on page 374

The name of the share is presented in the upper left corner to the right of the project panel. The first component of the name is the containing project, and clicking on the project name will navigate to the project details. The name of the share can also be changed by clicking on the share name and entering new text into the input. You will be asked to confirm this action, as it will require disconnecting active clients of the share.

### Share Usage Statistics

On the left side of the view (beneath the project panel when expanded) is a table explaining the current space usage statistics. These statistics are either for a particular share (when editing a share) or for the pool as a whole (when looking at the list of shares). If any properties are zero, then they are excluded from the table. The following usage statistics are shown:

- **Available space** - This statistic is implicitly shown as the capacity in terms of capacity percentage in the title. The available space reflects any quotas on the share or project, or the absolute capacity of the pool. The number shown here is the sum of the total space used and the amount of available space.

- **Referenced Data** - The amount of data referenced by the data. This includes all filesystem data or LUN blocks, in addition to requisite metadata. With compression, this value may be much less than the logical size of the data contained within the share. If the share is a clone of a snapshot, this value may be less than the physical storage it could theoretically include, and may be zero.

- **Snapshot Data** - The amount of space used by all snapshots of the share, including any project snapshots. This size is not equal to the sum of unique space consumed by all
snapshots. Blocks that are referenced by multiple snapshots are not included in the per-snapshot usage statistics, but will show up in the share's snapshot data total.

- **Unused Reservation** - If a filesystem has a reservation set, this value indicates the amount of remaining space that is reserved for the filesystem. This value is not set for LUNs. The appliance prevents other shares from consuming this space, guaranteeing the filesystem enough space. If the reservation does not include snapshots, then there must be enough space when taking a snapshot for the entire snapshot to be overwritten. For more information on reservations, see “Working with the Shares > Shares > General BUI Page” on page 309.

- **Total Space** - The sum of referenced data, snapshot data, and unused reservation.

## Share Static Properties

The left side of the shares view also shows static (create time) properties when editing a particular share. These properties are set at creation time, and cannot be modified once they are set. The following static properties are shown:

- **Compression Ratio** - If compression is enabled, this shows the compression ratio currently achieved for the share. This is expressed as a multiplier. For example, a compression of 2x means that the data is consuming half as much space as the uncompressed contents. For more information on compression and the available algorithms, see “Working with the Shares > Shares > General BUI Page” on page 309.

- **Case Sensitivity** - Controls whether directory lookups are case-sensitive or case-insensitive. It supports the following options:

<table>
<thead>
<tr>
<th>BUI Value</th>
<th>CLI Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixed</td>
<td>mixed</td>
<td>Case sensitivity depends on the protocol being used. For NFS, FTP, and HTTP, lookups are case-sensitive. For SMB, lookups are case-insensitive. This is default, and prioritizes conformance of the various protocols over cross-protocol consistency. When using this mode, it's possible to create files that are distinct over case-sensitive protocols, but clash when accessed over SMB. In this situation, the SMB server will create a &quot;mangled&quot; version of the conflicts that uniquely identify the filename.</td>
</tr>
<tr>
<td>Insensitive</td>
<td>insensitive</td>
<td>All lookups are case-insensitive, even over protocols (such as NFS) that are traditionally case-sensitive. This can cause confusion for clients of these protocols, but</td>
</tr>
</tbody>
</table>
Reject non UTF-8 - This setting enforces UTF-8 encoding for all files and directories. When set, attempts to create a file or directory with an invalid UTF-8 encoding will fail. This only affects NFSv3, where the encoding is not defined by the standard. NFSv4 always uses UTF-8, and SMB negotiates the appropriate encoding. This setting should normally be "on", or else SMB (which must know the encoding in order to do case sensitive comparisons, among other things) will be unable to decode filenames that are created with and invalid UTF-8 encoding. This setting should only be set to "off" in pre-existing NFSv3 deployments where clients are configured to use different encodings. Enabling SMB or NFSv4 when this property is set to "off" can yield undefined results if a NFSv3 client creates a file or directory that is not a valid UTF-8 encoding. This property must be set to "on" if the normalization property is set to anything other than "none".

Normalization - This setting controls what unicode normalization, if any, is performed on filesystems and directories. Unicode supports the ability to have the same logical name represented by different encodings. Without normalization, the on-disk name stored will be different, and lookups using one of the alternative forms will fail depending on how the file was created and how it is accessed. If this property is set to anything other than "none" (the default), the "Reject non UTF-8" property must also be set to "on". For more information on how normalization works, and how the different forms work, see the Wikipedia entry on unicode normalization.
<table>
<thead>
<tr>
<th>BUI Value</th>
<th>CLI Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form KC</td>
<td>formKC</td>
<td>Normalization Form Compatibility Composition (NFKC) - Characters are decomposed by compatibility equivalence, then recomposed by canonical equivalence.</td>
</tr>
<tr>
<td>Form KD</td>
<td>formKD</td>
<td>Normalization Form Compatibility Decomposition (NFKD) - Characters are decomposed by compatibility equivalence.</td>
</tr>
</tbody>
</table>

- **Volume Block Size** - The native block size for LUNs. This can be any power of 2 from 512 bytes to 1M, and the default is 8K.
- **Origin** - If this is a clone, this is the name of the snapshot from which it was cloned.
- **Data Migration Source** - If set, then this filesystem is actively shadowing an existing filesystem, either locally or over NFS. For more information about data migration, see “Working with Shadow Migration” on page 374.

### Using the Shares Project Panel

In the BUI, the set of available projects is always available via the project panel at the left side of the view. To expand or collapse the project panel, click the triangle by the "Projects" title bar.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Expand Icon" /></td>
<td>Expand project panel</td>
</tr>
<tr>
<td><img src="image2.png" alt="Collapse Icon" /></td>
<td>Collapse project panel</td>
</tr>
</tbody>
</table>

Selecting a project from the panel will navigate to the Projects view for the selected project. This project panel will also expand automatically when the move tool is clicked on a row within the share list. You can then drag and drop the share to move it between projects. The project panel also allows a shortcut for creating new projects, and reverting to the list of shares across all projects. Clicking the "All" text is equivalent to selecting the "Shares" item in the navigation bar.

The project panel is a convenience for systems with a relatively small number of projects. It is not designed to be the primary interface for managing a large number of projects. For this task, use the Projects view. See “Working with Projects” on page 382.
Creating a Share (BUI)

Shares, which are filesystems and LUNs are grouped into projects. A project defines common administrative control points for managing shares. Shares within a project can share common settings, and quotas can be enforced at the project level in addition to the share level. When you create a share, it inherits properties from its parent project. Some of those properties, such as encryption, are immutable.

For more information about projects, see “Working with Projects” on page 382.

To create a share:

1. **Go to Shares > Shares. From here, you can view existing shares per project or across all projects.**

2. **Select Filesystems or LUNs.**

3. **Click the Add item icon 🎨.**
   The Create Filesystem or Create LUN dialog box is displayed.

4. **In the dialog box, select or type the parameters you want to use.**
   For information about the filesystem or LUN parameters, see “Create Filesystem Parameters” on page 301 or “Create LUN Parameters” on page 303.

Create Filesystem Parameters

The following parameters are available for creating a filesystem.
Creating a Share (BUI)

**FIGURE 42** Create a Filesystem

- **Project** - Defines which project the filesystem uses to inherit parameter settings. You can also select the default project.
- **Name** - Name to identify the filesystem.
- **Data migration source** - Location of the source if you are migrating data
- **User** - Root directory owner, which can be specified as a user ID or user name.
- **Group** - Root directory group, which can be specified as a group ID or group name.
- **Permissions** - Sets the root permissions for the filesystem.
- **Inherit mountpoint** - Inherits the mountpoint from the project when selected. If not selected, enter a mountpoint, for example /export/foo, in the field below.
■ **Reject non UTF-8** - Enforces UTF-8 encoding for all files and directories when selected. Attempts to create a file or directory with an invalid UTF-8 encoding will fail. For more information about Reject non UTF-8, see “Share Static Properties” on page 298.

■ **Case sensitivity** - Sets the case sensitivity to sensitive, insensitive, or mixed. For more information, see “Share Static Properties” on page 298.

■ **Normalization** - Defines the unicode normalization. For more information about normalization, see “Share Static Properties” on page 298.

■ **Encryption** - Defines the encryption type. To inherit the encryption key from the project, click Local or OKM and select a key. To use encryption you must configure it first, see “Working with Data Encryption” on page 394. For information about encryption properties, see “Encryption Properties” on page 409. To understand the key values, see “Understanding Encryption Key Values” on page 411.

### Create LUN Parameters

The following parameters are available for creating a LUN.
FIGURE 43  Create LUN Parameters

- **Project** - Defines which project the LUN uses to inherit parameter settings. You can also select the default project.
- **Name** - Name to identify the LUN.
- **Volume size** - Maximum volume size and unit of measurement.
- **Thin provisioned** - Uses only the amount of space physically consumed by data when selected.
- **Volume block size** - Native block size for the LUN. This can be any power of 2 from 512 bytes to 1M, and the default is 8K.
- **Online** - On by default.
- **Target group** - Group of targets used when exporting this LUN.
- **Initiator group(s)** - Group(s) of initiators that can access this LUN.
- **LU number** - Sets the logical unit number to zero or automatically assigns it.
- **Encryption** - Defines the encryption type. To inherit the encryption key from the project, click Local or OKM and select a key. To use encryption you must configure it first, see “Working with Data Encryption” on page 394. For information about encryption properties, see “Encryption Properties” on page 409. To understand the key values, see “Understanding Encryption Key Values” on page 411.

## Selecting Shares (CLI)

The shares CLI is under `shares`

You must first select a project (including the default project) before selecting a share.

```
clownfish:> shares
clownfish:shares> select default
clownfish:shares default> select foo
clownfish:shares default/foo> get
Properties:
  aclinherit = restricted (inherited)
  aclmode = discard (inherited)
  atime = true (inherited)
  checksum = fletcher4 (inherited)
  compression = off (inherited)
  dedup = false (inherited)
  copies = 1 (inherited)
  logbias = latency (inherited)
  mountpoint = /export/foo (inherited)
  quota = 0 (inherited)
  readonly = false (inherited)
  recordsize = 128K (inherited)
  reservation = 0 (inherited)
  rstchown = true (inherited)
  secondarycache = all (inherited)
  nbmand = false (inherited)
  sharesmb = off (inherited)
  sharenfs = on (inherited)
  snapdir = hidden (inherited)
  vskan = false (inherited)
  sharedav = off (inherited)
  shareftp = off (inherited)
  sharesftp = off (inherited)
  sharetftp = off (inherited)
  root group = other (default)
  root permissions = 700 (default)
  root user = nobody (default)
  exported = true (inherited)
  casesensitivity = (default)
  normalization = (default)
  shadow = (default)
  utf8only = (default)
```
defaultuserquota = (default)
defaultgroupquota = (default)
encryption = (default)
snaplabel = (default)
quota_snap = (default)
reservation_snap = (default)
canonical_name = (default)
keyname = (default)
keystore = (default)
nodestroy = (default)

Share Operations (CLI)

A share is created by selecting the project and issuing the filesystem or lun command. The properties can be modified as needed before committing the changes:

clownfish:shares default> filesystem foo
clownfish:shares default/foo (uncommitted)> get
   aclinherit = restricted (inherited)
   aclmode = discard (inherited)
   atime = true (inherited)
   checksum = fletcher4 (inherited)
   compression = off (inherited)
   copies = 1 (inherited)
   mountpoint = /export/foo (inherited)
   quota = 0 (inherited)
   readonly = false (inherited)
   recordsize = 128K (inherited)
   reservation = 0 (inherited)
   secondarycache = all (inherited)
   nbmand = false (inherited)
   shareofs = off (inherited)
   sharefs = on (inherited)
   snapdir = hidden (inherited)
   snaplabel = project1:share1
   vscan = false (inherited)
   sharedav = off (inherited)
   shareftp = off (inherited)
   root_group = other (default)
   root_permissions = 700 (default)
   root_user = nobody (default)
   casesensitivity = (default)
   normalization = (default)
   utf8only = (default)
   quota_snap = (default)
   reservation_snap = (default)
   custom:int = (default)
   custom:string = (default)
   custom:email = (default)
clownfish:shares default/foo (uncommitted)> set shareofs=off
   shareofs = off (uncommitted)
A share can be destroyed using the `destroy` command from the share context:

```
clownfish:shares default/foo> destroy
This will destroy all data in "foo"! Are you sure? (Y/N)
```

A share can be renamed from the project context using the `rename` command:

```
clownfish:shares default> rename foo bar
```

A share can be moved between projects from the project context using the `move` command:

```
clownfish:shares default> move foo home
```

User and group usage and quotas can be managed through the `users` or `groups` commands after selecting the particular project or share. For more information on how to manage user and group quotas, see the “Space Management” on page 287 section.

## Shares CLI Properties

The following properties are available in the CLI, with their equivalent in the BUI. Properties can be set using the standard CLI commands `get` and `set`. In addition, properties can be inherited from the parent project by using the `unset` command.

<table>
<thead>
<tr>
<th>TABLE 125</th>
<th>Shares &gt; Shares CLI Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CLI Name</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>aclinherit</td>
<td>inherited</td>
</tr>
<tr>
<td>aclmode</td>
<td>inherited</td>
</tr>
<tr>
<td>atime</td>
<td>inherited</td>
</tr>
<tr>
<td>casesensitivity</td>
<td>create time</td>
</tr>
<tr>
<td>checksum</td>
<td>inherited</td>
</tr>
<tr>
<td>compression</td>
<td>inherited</td>
</tr>
<tr>
<td>compressratio</td>
<td>read-only</td>
</tr>
<tr>
<td>copies</td>
<td>inherited</td>
</tr>
<tr>
<td>creation</td>
<td>read-only</td>
</tr>
<tr>
<td>dedup</td>
<td>inherited</td>
</tr>
<tr>
<td>encryption</td>
<td>create time</td>
</tr>
<tr>
<td>CLI Name</td>
<td>Type</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>exported</td>
<td>inherited, replication packages only</td>
</tr>
<tr>
<td>fixednumber</td>
<td>LUN local</td>
</tr>
<tr>
<td>initiatorgroup</td>
<td>LUN local</td>
</tr>
<tr>
<td>keyname</td>
<td>static</td>
</tr>
<tr>
<td>keystatus</td>
<td>static when inherited from the project</td>
</tr>
<tr>
<td>keystore</td>
<td>static when inherited from the project</td>
</tr>
<tr>
<td>logbias</td>
<td>inherited</td>
</tr>
<tr>
<td>lunumber</td>
<td>LUN local</td>
</tr>
<tr>
<td>lunguid</td>
<td>read-only, LUN local</td>
</tr>
<tr>
<td>mountpoint</td>
<td>inherited</td>
</tr>
<tr>
<td>nbmand</td>
<td>inherited</td>
</tr>
<tr>
<td>nodelay</td>
<td>inherited</td>
</tr>
<tr>
<td>normalization</td>
<td>create time</td>
</tr>
<tr>
<td>origin</td>
<td>read-only</td>
</tr>
<tr>
<td>quota</td>
<td>space management</td>
</tr>
<tr>
<td>quota_snap</td>
<td>space management</td>
</tr>
<tr>
<td>readonly</td>
<td>inherited</td>
</tr>
<tr>
<td>recordsize</td>
<td>inherited</td>
</tr>
<tr>
<td>reservation</td>
<td>space management</td>
</tr>
<tr>
<td>reservation_snap</td>
<td>space management</td>
</tr>
<tr>
<td>root_group</td>
<td>filesystem local</td>
</tr>
<tr>
<td>root_permissions</td>
<td>filesystem local</td>
</tr>
<tr>
<td>root_user</td>
<td>filesystem local</td>
</tr>
<tr>
<td>rstchown</td>
<td>inherited</td>
</tr>
<tr>
<td>secondary_cache</td>
<td>inherited</td>
</tr>
<tr>
<td>shadow</td>
<td>create time</td>
</tr>
<tr>
<td>sharedav</td>
<td>inherited</td>
</tr>
<tr>
<td>shareftp</td>
<td>inherited</td>
</tr>
<tr>
<td>sharernfs</td>
<td>inherited</td>
</tr>
<tr>
<td>sharesmb</td>
<td>inherited</td>
</tr>
<tr>
<td>snapdir</td>
<td>inherited</td>
</tr>
<tr>
<td>snaplabel</td>
<td>inherited</td>
</tr>
</tbody>
</table>
Working with Shares > Shares > General BUI Page

This section of the BUI controls overall settings for the share that are independent of any particular protocol and are not related to access control or snapshots. While the CLI groups all properties in a single list, this section describes the behavior of the properties in both contexts.

These are standard properties that can either be inherited from the project or explicitly set on the share. The BUI only allows the properties to be inherited all at once, while the CLI allows for individual properties to be inherited.

Space within a storage pool is shared between all shares. Filesystems can grow or shrink dynamically as needed, though it is also possible to enforce space restrictions on a per-share basis. Quotas and reservations can be enforced on a per-filesystem basis. Quotas can also be enforced per-user and per-group. For more information on managing space usage for filesystems, including quotas and reservations, see “Space Management” on page 287.

The logical size of the LUN as exported over iSCSI. This property is only valid for LUNs. This property controls the size of the LUN. By default, LUNs reserve enough space to completely fill the volume. Changing the size of a LUN while actively exported to clients may yield undefined results. It may require clients to reconnect and/or cause data corruption on the filesystem on top of the LUN. Check best practices for your particular iSCSI client before attempting this operation.

Controls whether space is reserved for the volume. This property is only valid for LUNs. By default, a LUN reserves exactly enough space to completely fill the volume. This ensures that clients will not get out-of-space errors at inopportune times. This property allows the volume size to exceed the amount of available space. When set, the LUN will consume only the space that has been written to the LUN. While this allows for thin provisioning of LUNs, most filesystems do not expect to get "out of space" from underlying devices, and if the share runs out of space, it may cause instability and/or data corruption on clients.
When not set, the volume size behaves like a reservation excluding snapshots. It therefore has the same pathologies, including failure to take snapshots if the snapshot could theoretically diverge to the point of exceeding the amount of available space.

**Shares > Shares > General BUI Page Settings**

The following list describes the Shares > Shares > General BUI page settings.

- **Read only** - Controls whether the filesystem contents are read only. This property is only valid for filesystems. The contents of a read only filesystem cannot be modified, regardless of any protocol settings. This setting does not affect the ability to rename, destroy, or change properties of the filesystem. In addition, when a filesystem is read only, access control properties cannot be altered, because they require modifying the attributes of the root directory of the filesystem.

- **Update access time on read** - Controls whether the access time for files is updated on read. This property is only valid for filesystems. POSIX standards require that the access time for a file properly reflect the last time it was read. This requires issuing writes to the underlying filesystem even for a mostly read only workload. For working sets consisting primarily of reads over a large number of files, turning off this property may yield performance improvements at the expense of standards conformance. These updates happen asynchronously and are grouped together, so its effect should not be visible except under heavy load.

- **Non-blocking mandatory locking** - Controls whether SMB locking semantics are enforced over POSIX semantics. This property is only valid for filesystems. By default, filesystems implement file behavior according to POSIX standards. These standards are fundamentally incompatible with the behavior required by the SMB protocol. For shares where the primary protocol is SMB, this option should always be enabled. Changing this property requires all clients to be disconnected and reconnect.

- **Data deduplication** - Controls whether duplicate copies of data are eliminated. Deduplication is synchronous, pool-wide, block-based, and can be enabled on a per project or share basis. Enable it by selecting the Data Deduplication checkbox on the general properties screen for projects or shares. The deduplication ratio will appear in the usage area of the Status Dashboard. Data written with deduplication enabled is entered into the deduplication table indexed by the data checksum. Deduplication forces the use of the cryptographically strong SHA-256 checksum. Subsequent writes will identify duplicate data and retain only the existing copy on disk. Deduplication can only happen between blocks of the same size, data written with the same record size. As always, for best results set the record size to that of the application using the data; for streaming workloads use a large record size. If your data doesn't contain any duplicates, enabling Data Deduplication will add overhead (a more CPU-intensive checksum and on-disk deduplication table entries) without providing any benefit. If your data does contain duplicates, enabling Data Deduplication will both save space by storing only one copy of a given block regardless of how many times it occurs. Deduplication necessarily will impact performance in that the checksum is more expensive to compute and the metadata of the deduplication table must be accessed and maintained.
**Data compression** - Controls whether data is compressed before being written to disk. Shares can optionally compress data before writing to the storage pool. This allows for much greater storage utilization at the expense of increased CPU utilization. By default, no compression is done. If the compression does not yield a minimum space savings, it is not committed to disk to avoid unnecessary decompression when reading back the data. Before choosing a compression algorithm, it is recommended that you perform any necessary performance tests and measure the achieved compression ratio.

<table>
<thead>
<tr>
<th>BUI value</th>
<th>CLI value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>off</td>
<td>No compression is done</td>
</tr>
<tr>
<td>LZJB (Fastest)</td>
<td>lzjb</td>
<td>A simple run-length encoding that only works for sufficiently simple inputs, but doesn't consume much CPU.</td>
</tr>
<tr>
<td>GZIP-2 (Fast)</td>
<td>gzip-2</td>
<td>A lightweight version of the gzip compression algorithm.</td>
</tr>
<tr>
<td>GZIP (Default)</td>
<td>gzip</td>
<td>The standard gzip compression algorithm.</td>
</tr>
<tr>
<td>GZIP-9 (Best Compression)</td>
<td>gzip-9</td>
<td>Highest achievable compression using gzip. This consumes a significant amount of CPU and can often yield only marginal gains.</td>
</tr>
</tbody>
</table>

**Checksum** - Controls the checksum used for data blocks. On the appliance, all data is checksummed on disk, and in such a way to avoid traditional pitfalls (phantom reads and write in particular). This allows the system to detect invalid data returned from the devices. The default checksum (fletcher4) is sufficient for normal operation, but paranoid users can increase the checksum strength at the expense of additional CPU load. Metadata is always checksummed using the same algorithm, so this only affects user data (files or LUN blocks).

<table>
<thead>
<tr>
<th>BUI value</th>
<th>CLI value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fletcher 2 (Legacy)</td>
<td>fletcher2</td>
<td>16-bit fletcher checksum</td>
</tr>
<tr>
<td>Fletcher 4 (Standard)</td>
<td>fletcher4</td>
<td>32-bit fletcher checksum</td>
</tr>
<tr>
<td>SHA-256 (Extra Strong)</td>
<td>sha256</td>
<td>SHA-256 checksum</td>
</tr>
</tbody>
</table>

**Cache Device Usage** - Controls whether cache devices are used for the share. By default, all datasets make use of any cache devices on the system. Cache devices are configured as part of the storage pool and provide an extra layer of caching for faster tiered access. For more information on cache devices, see “Storage Configuration” on page 77. This property is independent of whether there are any cache devices currently configured in the storage pool. For example, it is possible to have this property set to “all” even if there are no cache devices present. If any such devices are added in the future, the share will automatically take advantage of the additional performance. This property does not affect use of the primary (DRAM) cache.
BUI value | CLI value | Description
---|---|---
All data and metadata | all | All normal file or LUN data is cached, as well as any metadata.
Metadata only | metadata | Only metadata is kept on cache devices. This allows for rapid traversal of directory structures, but retrieving file contents may require reading from the data devices.
Do not use cache devices | none | No data in this share is cached on the cache device. Data is only cached in the primary cache or stored on data devices.

**Synchronous Write Bias** - This setting controls the behavior when servicing synchronous writes. By default, the system optimizes synchronous writes for latency, which leverages the log devices to provide fast response times. In a system with multiple disjointed filesystems, this can cause contention on the log devices that can increase latency across all consumers. Even with multiple filesystems requesting synchronous semantics, it may be the case that some filesystems are more latency-sensitive than others.

A common case is a database that has a separate log. The log is extremely latency sensitive, and while the database itself also requires synchronous semantics, it is heavier bandwidth and not latency sensitive. In this environment, setting this property to 'throughput' on the main database while leaving the log filesystem as 'latency' can result in significant performance improvements. This setting will change behavior even when no log devices are present, though the effects may be less dramatic.

The Synchronous write bias setting can be bypassed by the Oracle Intelligent Storage Protocol. Instead of using the write bias defined in the file system, the Oracle Intelligent Storage Protocol can use the write bias value provided by the Oracle Database NFSv4 client. The write bias value sent by the Oracle Database NFSv4 client is used only for that write request.

<table>
<thead>
<tr>
<th>BUI value</th>
<th>CLI value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>latency</td>
<td>Synchronous writes are optimized for latency, leveraging the dedicated log device(s), if any.</td>
</tr>
<tr>
<td>Throughput</td>
<td>throughput</td>
<td>Synchronous writes are optimized for throughput. Data is written to the primary data disks instead of the log device(s), and the writes are performed in a way that optimizes for total bandwidth of the system.</td>
</tr>
</tbody>
</table>

**Database Record Size** - Specifies a suggested block size for files in the file system. This property is only valid for filesystems and is designed for use with database workloads that
access files in fixed-size records. The system automatically tunes block sizes according to internal algorithms optimized for typical access patterns. For databases that create very large files but access them in small random chunks, these algorithms may be suboptimal. Specifying a record size greater than or equal to the record size of the database can result in significant performance gains. Use of this property for general purpose file systems is strongly discouraged, and may adversely affect performance.

The default record size is 128 KB. The size specified must be a power of two greater than or equal to 512 and less than or equal to 1 MB. Changing the file system's record size affects only files created afterward; existing files and received data are unaffected. If block sizes greater than 128K are used for projects or shares, replication of those projects or shares to systems that don't support large block sizes will fail.

The Database record size setting can be bypassed by the Oracle Intelligent Storage Protocol. Instead of using the record size defined in the file system the Oracle Intelligent Storage Protocol can use the block size value provided by the Oracle Database NFSv4 client. The block size provided by the Oracle Database NFSv4 client can only be applied when creating a new database files or table. Block sizes of existing files and tables will not be changed. For more information, see “Oracle Intelligent Storage Protocol” on page 445.

- **Additional Replication** - Controls number of copies stored of each block, above and beyond any redundancy of the storage pool. Metadata is always stored with multiple copies, but this property allows the same behavior to be applied to data blocks. The storage pool attempts to store these extra blocks on different devices, but it is not guaranteed. In addition, a storage pool cannot be imported if a complete logical device (RAID stripe, mirrored pair, etc.) is lost. This property is not a replacement for proper replication in the storage pool, but can be reassuring for paranoid administrators.

<table>
<thead>
<tr>
<th>BUI value</th>
<th>CLI value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal (Single Copy)</td>
<td>1</td>
<td>Default behavior. Store a single copy of data blocks.</td>
</tr>
<tr>
<td>Two Copies</td>
<td>2</td>
<td>Store two copies of every data block.</td>
</tr>
<tr>
<td>Three Copies</td>
<td>3</td>
<td>Store three copies of every data block.</td>
</tr>
</tbody>
</table>

- **Virus Scan** - Controls whether this filesystem is scanned for viruses. This property is only valid for filesystems. This property setting is independent of the state of the virus scan service. Even if the Virus Scan service is enabled, filesystem scanning must be explicitly enabled using this property. Similarly, virus scanning can be enabled for a particular share even if the service itself is off. For more information about configuration virus scanning, see “Virus Scan” on page 220.

- **Prevent Destruction** - When set, the share or project cannot be destroyed. This includes destroying a share through dependent clones, destroying a share within a project, or destroying a replication package. However, it does not affect shares destroyed through replication updates. If a share is destroyed on an appliance that is the source for replication, the corresponding share on the target will be destroyed, even if this property is
set. To destroy the share, the property must first be explicitly turned off as a separate step. This property is off by default.

- **Restrict Ownership Change** - By default, ownership of files cannot be changed except by a root user (on a suitable client with a root-enabled export). This property can be turned off on a per-filesystem or per-project basis by turning off this property. When off, file ownership can be changed by the owner of the file or directory, effectively allowing users to "give away" their own files. When ownership is changed, any setuid or setgid bits are stripped, preventing users from escalating privileges through this operation.

- **Custom Properties** - Custom properties can be added as needed to attach user-defined tags to projects and shares. For more information, see “Working with Schemas” on page 391.

---

### Understanding the Shares > Shares > Protocols BUI Page

Each share has protocol-specific properties which define the behavior of different protocols for that share. These properties may be defined for each share or inherited from a share's project.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFS</td>
<td>twofish:/export/fs0</td>
</tr>
<tr>
<td>SMB</td>
<td>\twofish\fs0</td>
</tr>
<tr>
<td>HTTP</td>
<td>//twofish/shares/export/fs0/</td>
</tr>
<tr>
<td>FTP</td>
<td>ftp://twofish/export/fs0/</td>
</tr>
<tr>
<td>SFTP</td>
<td>/export/fs0/</td>
</tr>
</tbody>
</table>

For iSCSI, initiators can discover the target through one of the mechanisms described in “Configuring Storage Area Network” on page 99.

To work with the Shares > Shares > Protocols BUI page, see the following sections:

- “Share Protocols - NFS” on page 315
- “Configuring Share NFS Protocols using the CLI” on page 316
- “Configuring Share NFS Protocol Security Modes” on page 317
- “Configuring Share NFS Protocol Character Set Encodings” on page 318
- “Shares Protocols - SMB” on page 319
- “Shares Protocols - iSCSI” on page 320
- “Shares Protocols - HTTP” on page 321
- “Shares Protocols - FTP” on page 321
- “Shares Protocols - SFTP” on page 321
# Share Protocols - NFS

The following table shows the Share Protocols - NFS properties.

<table>
<thead>
<tr>
<th>TABLE 126</th>
<th>Share Protocols - NFS Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BUI Property</strong></td>
<td><strong>CLI Property</strong></td>
</tr>
<tr>
<td>Share mode</td>
<td>off/ro/rw</td>
</tr>
<tr>
<td>Disable setuid/setgid file creation</td>
<td>nosuid</td>
</tr>
<tr>
<td>Prevent clients from mounting subdirectories</td>
<td>nosub</td>
</tr>
<tr>
<td>Anonymous user mapping</td>
<td>anon</td>
</tr>
<tr>
<td>Character encoding</td>
<td>See below</td>
</tr>
<tr>
<td>Security mode</td>
<td>See below</td>
</tr>
</tbody>
</table>

Exceptions to the overall sharing modes may be defined for clients or collections of clients. When a client attempts access, its access will be granted according to the first exception in the list that matches the client; or, if no such exception exists, according to the global share modes defined above. These client collections may be defined using one of three types:

<table>
<thead>
<tr>
<th>TABLE 127</th>
<th>Client Collection Types</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td><strong>CLI Prefix</strong></td>
</tr>
<tr>
<td>Host(FQDN) or Netgroup</td>
<td>none</td>
</tr>
<tr>
<td>Type</td>
<td>CLI Prefix</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>DNS Domain</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td>@</td>
</tr>
</tbody>
</table>

For each specified client or collection of clients, you will then express two parameters: whether the client shall be permitted read-only or read-write access to the share, and whether the root user on the client shall be treated as the root user (if selected) or the unknown user.

If netgroups are used, they will be resolved from NIS (if enabled) and then from LDAP (if enabled). If LDAP is used, the netgroups must be found at the default location, ou=Netgroup, (Base DN), and must use the standard schema. The username component of a netgroup entry typically has no effect on NFS; only the hostname is significant. Hostnames contained in netgroups must be canonical and, if resolved using DNS, fully qualified. That is, the NFS subsystem will attempt to verify that the IP address of the requesting client resolves to a canonical hostname that matches either the specified FQDN or one of the members of one of the specified netgroups. This match must be exact, including any domain components; otherwise, the exception will not match and the next exception will be tried. For more information on hostname resolution, see “DNS” on page 239. Management of netgroups can be complex; consider using IP subnet rules or DNS domain rules instead where possible.

As of the 2013.1.0 software release, Unix client users may belong to a maximum of 1024 groups without any performance degradation. Prior releases supported up to 16 groups per Unix client user.

### Configuring Share NFS Protocols using the CLI

In the CLI, all NFS share modes and exceptions are specified using a single options string for the "sharenfs" property. This string is a comma-separated list of values from the tables above. It should begin with one of "ro", "rw", or "off", as an analogue to the global share modes described for the BUI. For example,

```bash
set sharenfs=ro
```

sets the share mode for all clients to read-only. The root users on all clients will access the files on the share as if they were the generic "nobody" user.
Either or both of the "nosuid" and "anon" options may also be appended. Remember that in the CLI, property values containing the "=" character must be quoted. Therefore, to define the mapping of all unknown users to the uid 153762, you might specify:

```
set sharenfs="ro,anon=153762"
```

Additional exceptions can be specified by appending text of the form "option=collection", where "option" is one of "ro", "rw", and "root", defining the type of access to be granted to the client collection. The collection is specified by the prefix character from the table above and either a DNS hostname/domain name or CIDR network number. For example, to grant read-write access to all hosts in the sf.example.com domain and root access to those in the 192.168.44.0/24 network, you might use:

```
set sharenfs="ro,anon=153762,rw=.sf.example.com,root=@192.168.44.0/24"
```

Netgroup names can be used anywhere an individual fully-qualified hostname can be used. For example, you can permit read-write access to the "engineering" netgroup as follows:

```
set sharenfs="ro,rw=engineering"
```

Security modes are specified by appending text in the form "option=mode" where option is "sec" and mode is one of "sys", "krb5", "krb5:krb5i", or "krb5:krb5i:krb5p".

```
set sharenfs="sec=krb5"
```

### Configuring Share NFS Protocol Security Modes

Security modes are set on per-share basis and can have performance impact. The following table describes the Kerberos security settings.

<table>
<thead>
<tr>
<th>Kerberos Security Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>krb5</td>
<td>End-user authentication through Kerberos V5</td>
</tr>
<tr>
<td>krb5i</td>
<td>krb5 plus integrity protection (data packets are tamper proof)</td>
</tr>
<tr>
<td>krb5p</td>
<td>krb5i plus privacy protection (data packets are tamper proof and encrypted)</td>
</tr>
</tbody>
</table>

Combinations of Kerberos flavors may be specified in the security mode setting. The combination security modes let clients mount with any Kerberos flavor listed.
### TABLE 129 Security Mode Settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>sys</td>
<td>System Authentication</td>
</tr>
<tr>
<td>krb5</td>
<td>Kerberos v5 only - Clients must mount using this flavor.</td>
</tr>
<tr>
<td>krb5:krb5i</td>
<td>Kerberos v5, with integrity - Clients may mount using any flavor listed.</td>
</tr>
<tr>
<td>krb5i</td>
<td>Kerberos v5 integrity only - Clients must mount using this flavor.</td>
</tr>
<tr>
<td>krb5:krb5i:krb5p</td>
<td>Kerberos v5, with integrity or privacy - Clients may mount using any flavor listed.</td>
</tr>
<tr>
<td>krb5p</td>
<td>Kerberos v5 privacy only - Clients must mount using this flavor.</td>
</tr>
</tbody>
</table>

For more information about NFS and Kerberos, see:
- [NFSv2 and NFSv3 Security](http://www.ietf.org/rfc/rfc2623.txt)
- [NFSv4 Protocol](http://www.ietf.org/rfc/rfc3530.txt)

### Configuring Share NFS Protocol Character Set Encodings

Normally, the character set encoding used for filename is unspecified. The NFSv3 and NFSv2 protocols don't specify the character set. NFSv4 is supposed to use UTF-8, but not all clients do and this restriction is not enforced by the server. If the UTF-8 only option is disabled for a share, these filenames are written verbatim to the filesystem without any knowledge of their encoding. This means that they can only be interpreted by clients using the same encoding. SMB, however, requires filenames to be stored as UTF-8 so that they can be interpreted on the server side. This makes it impossible to support arbitrary client encodings while still permitting access over SMB.

In order to support such configurations, the character set encoding can be set share-wide or on a per-client basis. The following character set encodings are supported:
- cp932
- euc-cn
- euc-jp
- euc-jpms
- euc-kr
- euc-tw
- iso8859-1
- iso8859-2
- iso8859-5
iso8859-6
iso8859-7
iso8859-8
iso8859-9
iso8859-13
iso8859-15
koi8-r
shift_jis

The default behavior is to leave the character set encoding unspecified (pass-through). The BUI allows the character set to be chosen through the standard exception list mechanism. In the CLI, each character set itself becomes an option with one or more hosts, with '*' indicating the share-wide setting. For example, the following:

```
set sharenfs="rw,euc-kr=*"
```

Will share the filesystem with 'euc-kr' as the default encoding. The following:

```
set sharenfs="rw,euc-kr=host1.domain.com,euc-jp=host2.domain.com"
```

Use the default encoding for all clients except 'host1' and 'host2', which will use 'euc-kr' and 'euc-jp', respectively. The format of the host lists follows that of other CLI NFS options.

Note that some NFS clients do not correctly support alternate locales; consult your NFS client documentation for details.

**Shares Protocols - SMB**

- **Resource name** - The name by which SMB clients refer to this share. The resource name "off" indicates no SMB client may access the share, and the resource name "on" indicates the share will be exported with the filesystem's name.

- **Enable Access-based Enumeration** - An option which, when enabled, performs access-based enumeration. Access-based enumeration filters directory entries based on the credentials of the client. When the client does not have access to a file or directory, that file will be omitted from the list of entries returned to the client. This option is not enabled by default.

- **Is a DFS Namespace** - A property which indicates whether this share is provisioned as a standalone DFS namespace.

- **Share-level ACL** - An ACL which is combined with the ACL of a file or directory in the share to determine the effective permissions for that file. By default, this ACL grants everyone full control. This ACL provides another layer of access control above the ACLs
on files and allows for more sophisticated access control configurations. This property may only be set once the filesystem has been exported by configuring the SMB resource name. If the filesystem is not exported over the SMB protocol, setting the share-level ACL has no effect.

No two SMB shares on the same system may share the same resource name. Resource names inherited from projects have special behavior, see “Working with Projects” on page 382 for details. Resource names must be less than 80 characters, and can contain any alphanumeric characters besides the following characters:

```
" / \ [ ] : | < > + ; , ? * =
```

When access-based enumeration is enabled, clients may see directory entries for files which they cannot open. Directory entries are filtered only when the client has no access to that file. For example, if a client attempts to open a file for read/write access but the ACL grants only read access, that open request will fail but that file will still be included in the list of entries.

**Shares Protocols - iSCSI**

- **Target group** - The targets over which this LUN is exported.
- **Initiator group(s)** - The initiators that can access this LUN. As of the 2013.1.0 software release, multiple initiator groups can be assigned to a LUN. When editing initiator groups, checking the PERSIST checkbox (the default) preserves the LUN number for the corresponding initiator group. If unchecked, the appliance may reassign the LUNs after a SAN configuration change or a reboot.
- **LU (logical unit) number** - As LUNs are associated with target and initiator groups, they are assigned unique logical unit numbers per target group and initiator pair. No two LUNs that are accessible by an initiator through a target group may share a logical unit number. This property controls whether a logical unit must have number zero or an automatically assigned number.
- **Operational status** - The operational status of this LUN. An offline LUN is inaccessible to initiators regardless of target or initiator configuration.
- **Write cache behavior** - This setting controls whether the LUN caches writes. With this setting off, all writes are synchronous and if no log device is available, write performance suffers significantly. Turning this setting on can therefore dramatically improve write performance, but can also result in data corruption on unexpected shutdown or failover unless the client application understands the semantics of a volatile write cache and properly flushes the cache when necessary. Consult your client application documentation before turning this on.
- **GUID** - A LUN's GUID is a globally unique, read-only identifier that identifies the SCSI device. This GUID remains consistent within different head nodes and replicated environments.
Shares Protocols - HTTP

TABLE 130 Shares - HTTP Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Share mode</td>
<td>The HTTP share mode for this filesystem. One of none, read only, or read/write.</td>
</tr>
</tbody>
</table>

Shares Protocols - FTP

TABLE 131 Shares - FTP Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Share mode</td>
<td>The FTP share mode for this filesystem. One of none, read only, or read/write.</td>
</tr>
</tbody>
</table>

Shares Protocols - SFTP

TABLE 132 Shares - SFTP Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Share mode</td>
<td>The SFTP share mode for this filesystem. One of none, read only, or read/write.</td>
</tr>
</tbody>
</table>

Working with the Shares > Shares > Access BUI Page

This view lets you set options to control ACL behavior as well as control access to the root directory of the filesystem. This view is only available for filesystems.

To work with the Shares > Shares > Access BUI page, see the following sections:

- “Shares - Root Directory Access” on page 321
- “Shares - Selecting Permissions” on page 322
- “Shares - ACL Behavior on Mode Change” on page 323
- “Shares - ACL Inheritance Behavior” on page 324
- “Shares - Root Directory ACL” on page 325

Shares - Root Directory Access

Controls basic access control for the root of the filesystem. These settings can be managed in-band via whatever protocols are being used, but they can also be specified here for convenience.
These properties cannot be changed on a read-only filesystem, as they require changing metadata for the root directory of the filesystem.

- **User** - The owner of the root directory. This can be specified as a user ID or user name. For more information on mapping Unix and Windows users, see “Identity Mapping” on page 233. For Unix-based NFS access, this can be changed from the client using the `chown` command.

- **Group** - The group of the root directory. This can be specified as a group ID or group name. For more information on mapping Unix and Windows groups, see “Identity Mapping” on page 233. For Unix-based NFS access, this can be changed from the client using the `chgrp` command.

- **Permissions** - Standard Unix permissions for the root directory. For Unix-based NFS access, this can be changed from the client using the `chmod` command. The permissions are divided into three types.

<table>
<thead>
<tr>
<th>Access type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>User that is the current owner of the directory.</td>
</tr>
<tr>
<td>Group</td>
<td>Group that is the current group of the directory.</td>
</tr>
<tr>
<td>Other</td>
<td>All other accesses.</td>
</tr>
</tbody>
</table>

For each access type, the following permissions can be granted.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>R</td>
</tr>
<tr>
<td>Write</td>
<td>W</td>
</tr>
<tr>
<td>Execute</td>
<td>X</td>
</tr>
</tbody>
</table>

Permission to list the contents of the directory.

Permission to create files in the directory.*

Permission to look up entries in the directory. If users have execute permissions but not read permissions, they can access files explicitly by name but not list the contents of the directory.

---

**Shares - Selecting Permissions**

In the BUI, selecting permissions is done by clicking on individual boxes. Alternatively, clicking on the label ("user," "group," or "other) will select (or deselect) all permissions within the label. In the CLI, permissions are specified as a standard Unix octal value, where each digit corresponds to (in order) user, group, and other. Each digit is the sum of read (4), write (2), and execute (1). So a permissions value of 743 would be the equivalent of user RWX, group R, other WX.
As an alternative to setting POSIX permission bits at share creation time, administrators may instead select the "Use Windows Default Permissions" option, which will apply an ACL as described in “Shares - Root Directory ACL” on page 325. This is a shortcut to simplify administration in environments that are exclusively or predominately managed by users with Windows backgrounds and is intended to provide behaviour similar to share creation on a Windows server.

- Child files within the directory can be deleted (same as the ACL D permission), unless the sticky bit is set on the directory, in which case the child files can be deleted only if requested by the owner of the file
- Times associated with a file or directory can be changed (same as the ACL A permission)
- Extended attributes can be created, and writes are allowed to the extended attributes directory (same as the ACL W permission)

### Shares - ACL Behavior on Mode Change

For information on ACLs and how they work, see “Shares - Root Directory ACL” on page 325.

When an ACL is modified via chmod(2) using the standard Unix user/group/other permissions, the simplified mode change request will interact with the existing ACL in different ways depending on the setting of this property.

<table>
<thead>
<tr>
<th>TABLE 133 Mode Change Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUI Value</td>
</tr>
<tr>
<td>Discard ACL</td>
</tr>
<tr>
<td>Mask ACL with mode</td>
</tr>
<tr>
<td>Do not change ACL</td>
</tr>
</tbody>
</table>
Shares - ACL Inheritance Behavior

When a new file or directory is created, it is possible to inherit existing ACL settings from the parent directory. This property controls how this inheritance works. These property settings usually only affect ACL entries that are flagged as inheritable - other entries are not propagated regardless of this property setting. However, all trivial ACL entries are inheritable when used with SMB. A trivial ACL represents the traditional Unix owner/group/other entries.

TABLE 134  ACL Inheritance Behavior Values

<table>
<thead>
<tr>
<th>BUI Value</th>
<th>CLI Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not inherit entries</td>
<td>discard</td>
<td>No ACL entries are inherited. The file or directory is created according to the client and protocol being used.</td>
</tr>
<tr>
<td>Only inherit deny entries</td>
<td>noallow</td>
<td>Only inheritable ACL entries specifying “deny” permissions are inherited.</td>
</tr>
<tr>
<td>Inherit all but &quot;write ACL&quot; and &quot;change owner&quot;</td>
<td>restricted</td>
<td>Removes the &quot;write_acl&quot; and &quot;write_owner&quot; permissions when the ACL entry is inherited, but otherwise leaves inheritable ACL entries untouched. This is the default.</td>
</tr>
<tr>
<td>Inherit all entries</td>
<td>passthrough</td>
<td>All inheritable ACL entries are inherited. The “passthrough” mode is typically used to cause all “data” files to be created with an identical mode in a directory tree. An administrator sets up ACL inheritance so that all files are created with a mode, such as 0664 or 0666.</td>
</tr>
<tr>
<td>Inherit all but &quot;execute&quot; when not specified</td>
<td>passthrough-x</td>
<td>Same as ‘passthrough’, except that the owner, group, and everyone ACL entries inherit the execute permission only if the file creation mode also requests the execute bit. The &quot;passthrough&quot; setting works as expected for data files, but you might want to optionally include the execute bit from the file creation mode into the inherited ACL. One example is an output file that is generated from tools, such as &quot;cc&quot; or &quot;gcc&quot;. If the inherited ACL doesn't include the execute bit, then the output executable from the compiler won't be executable until you use chmod(1) to change the file's permissions.</td>
</tr>
</tbody>
</table>

When using SMB to create a file in a directory with a trivial ACL, all ACL entries are inherited. As a result, the following behavior occurs:
■ Inheritance bits display differently when viewed in SMB or NFS. When viewing the ACL directory in SMB, inheritance bits are displayed. In NFS, inheritance bits are not displayed.

■ When a file is created in a directory using SMB, its ACL entries are shown as inherited; however, when viewed through NFS, the directory has no inheritable ACL entries.

■ If the ACL is changed so that it is no longer trivial, e.g., by adding an access control entry (ACE), this behavior does not occur.

■ If the ACL is modified using SMB, the resulting ACL will have the previously synthetic inheritance bits turned into real inheritance bits.

All of the above behavior is subject to change in a future release.

**Shares - Root Directory ACL**

Fine-grained access on files and directories is managed via Access Control Lists. An ACL describes what permissions are granted, if any, to specific users or groups. The appliance supports NFSv4-style ACLs, also accessible over SMB. POSIX draft ACLs (used by NFSv3) are not supported. Some trivial ACLs can be represented over NFSv3, but making complicated ACL changes may result in undefined behavior when accessed over NFSv3.

Like root directory access, this property only affects the root directory of the filesystem. ACLs can be controlled through in-band protocol management, but the BUI provides a way to set the ACL just for the root directory of the filesystem. There is no way to set the root directory ACL through the CLI. You can use in-band management tools if the BUI is not an option. Changing this ACL does not affect existing files and directories in the filesystem. Depending on the ACL inheritance behavior, these settings may or may not be inherited by newly created files and directories. However, all ACL entries are inherited when SMB is used to create a file in a directory with a trivial ACL.

An ACL is composed of any number of ACEs (access control entries). Each ACE describes a type/target, a mode, a set of permissions, and inheritance flags. ACEs are applied in order, starting at the beginning of the ACL, to determine whether a given action should be permitted. For information on in-band configuration ACLs through data protocols, consult the appropriate client documentation. The BUI interface for managing ACLs and the effect on the root directory are described here.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Current owner of the directory. If the owner is changed, this ACE will apply to the new owner.</td>
</tr>
<tr>
<td>Group</td>
<td>Current group of the directory. If the group is changed, this ACE will apply to the new group.</td>
</tr>
<tr>
<td>Everyone</td>
<td>Any user.</td>
</tr>
</tbody>
</table>
### Type

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Named User</td>
<td>User named by the 'target' field. The user can be specified as a user ID or a name resolvable by the current name service configuration.</td>
</tr>
<tr>
<td>Named Group</td>
<td>Group named by the 'target' field. The group can be specified as a group ID or a name resolvable by the current name service configuration.</td>
</tr>
</tbody>
</table>

#### TABLE 136  Share - ACL Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="allow_icon" alt="Allow" /></td>
<td>The permissions are explicitly granted to the ACE target.</td>
</tr>
<tr>
<td><img src="deny_icon" alt="Deny" /></td>
<td>The permissions are explicitly denied to the ACE target.</td>
</tr>
</tbody>
</table>

#### TABLE 137  Share - ACL Permissions

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read (r)</td>
<td>Permission to list the contents of a directory. When inherited by a file, permission to read the data of the file.</td>
</tr>
<tr>
<td>Read Data/List Directory (r)</td>
<td>Permission to list the contents of a directory. When inherited by a file, permission to read the data of the file.</td>
</tr>
<tr>
<td>Execute File/Traverse Directory (x)</td>
<td>Permission to traverse (lookup) entries in a directory. When inherited by a file, permission to execute the file.</td>
</tr>
<tr>
<td>Read Attributes (a)</td>
<td>Permission to read basic attributes (non-ACLs) of a file. Basic attributes are considered to be the stat level attributes, and allowing this permission means that the user can execute <code>ls</code> and <code>stat</code> equivalents.</td>
</tr>
<tr>
<td>Read Extended Attributes (R)</td>
<td>Permission to read the extended attributes of a file or do a lookup in the extended attributes directory.</td>
</tr>
<tr>
<td>Write (w)</td>
<td>Permission to add a new file to a directory. When inherited by a file, permission to modify a file's data anywhere in the file's offset range. This include the ability to grow the file or write to any arbitrary offset.</td>
</tr>
<tr>
<td>Write Data/Add File (w)</td>
<td>Permission to add a new file to a directory. When inherited by a file, permission to modify a file's data anywhere in the file's offset range. This include the ability to grow the file or write to any arbitrary offset.</td>
</tr>
<tr>
<td>Append Data/Add Subdirectory (p)</td>
<td>Permission to create a subdirectory within a directory. When inherited by a file, permission to modify the file's data, but only starting at the end of the file. This permission (when applied to files) is not currently supported.</td>
</tr>
<tr>
<td>Permission</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>(d) Delete</td>
<td>Permission to delete a file.</td>
</tr>
<tr>
<td>(D) Delete Child</td>
<td>Permission to delete a file within a directory. As of the 2011.1 software release, if the sticky bit is set, a child file can only be deleted by the file owner.</td>
</tr>
<tr>
<td>(A) Write Attributes</td>
<td>Permission to change the times associated with a file or directory.</td>
</tr>
<tr>
<td>(W) Write Extended Attributes</td>
<td>Permission to create extended attributes or write to the extended attributes directory.</td>
</tr>
<tr>
<td>(c) Read ACL/Permissions</td>
<td>Permission to read the ACL.</td>
</tr>
<tr>
<td>(C) Write ACL/Permissions</td>
<td>Permission to write the ACL or change the basic access modes.</td>
</tr>
<tr>
<td>(o) Change Owner</td>
<td>Permission to change the owner.</td>
</tr>
<tr>
<td>(f) Apply to Files</td>
<td>Inherit to all newly created files in a directory.</td>
</tr>
<tr>
<td>(d) Apply to Directories</td>
<td>Inherit to all newly created directories in a directory.</td>
</tr>
<tr>
<td>(i) Do not apply to self</td>
<td>The current ACE is not applied to the current directory, but does apply to children. This flag requires one of &quot;Apply to Files&quot; or &quot;Apply to Directories&quot; to be set.</td>
</tr>
<tr>
<td>(n) Do not apply past children</td>
<td>The current ACE should only be inherited one level of the tree, to immediate children. This flag requires one of &quot;Apply to Files&quot; or &quot;Apply to Directories&quot; to be set.</td>
</tr>
</tbody>
</table>

When the option to use Windows default permissions is used at share creation time, an ACL with the following three entries is created for the share's root directory:

### TABLE 138 Share Root Directory Entities

<table>
<thead>
<tr>
<th>Type</th>
<th>Action</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Allow</td>
<td>Full Control</td>
</tr>
<tr>
<td>Group</td>
<td>Allow</td>
<td>Read and Execute</td>
</tr>
<tr>
<td>Everyone</td>
<td>Allow</td>
<td>Read and Execute</td>
</tr>
</tbody>
</table>
Snapshots are read only copies of a filesystem at a given point of time. For more information on snapshots and how they work, see “Understanding Snapshots” on page 288.

The following lists the Snapshot properties.

- **.zfs/snapshot visible** - Filesystem snapshots can be accessed over data protocols at .zfs/snapshot in the root of the filesystem. This directory contains a list of all snapshots on the filesystem, and they can be accessed just like normal filesystem data (in read only mode). By default, the '.zfs' directory is not visible when listing directory contents, but can be accessed by explicitly looking it up. This prevents backup software from inadvertently backing up snapshots in addition to new data.

<table>
<thead>
<tr>
<th>BUI Value</th>
<th>CLI Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden</td>
<td>hidden</td>
<td>The .zfs directory is not visible when listing directory contents in the root of the filesystem. This is default.</td>
</tr>
<tr>
<td>Visible</td>
<td>visible</td>
<td>This .zfs directory appears like any other directory in the filesystem.</td>
</tr>
</tbody>
</table>

- **Scheduled Snapshot Label** - This optional property appends a user-defined label to each scheduled snapshot and is blank by default. The label can either be set for an individual share, or set for a project and inherited by its shares, but not both. Snapshot labels can help identify the project or share for which a snapshot was taken, for example "project1:share1" could indicate a scheduled snapshot taken on share1 within project1. Labels can be up to 35 alphanumeric characters and include special characters _ - . :  

To work with shares snapshots, see the following sections:

- “Listing Snapshots using the BUI” on page 329
- “Create a Project Level Snapshot using the BUI” on page 330
- “Create a Share/LUN Level Snapshot using the BUI” on page 330
- “Renaming a Snapshot using the BUI” on page 330
- “Destroying a Snapshot using the BUI” on page 331
- “Rolling back to a Snapshot using the BUI” on page 331
- “Cloning a Snapshot using the BUI” on page 331
- “Scheduling Snapshots using the BUI” on page 332
- “Manual Snapshots using the CLI” on page 333
- “Listing Snapshots using the CLI” on page 333
- “Taking Manual Snapshots using the CLI” on page 333
- “Renaming a Snapshot using the CLI” on page 334
- “Destroying a Snapshot using the CLI” on page 334
- “Rolling back to a Snapshot using the CLI” on page 334
Listing Snapshots using the BUI

Under the "snapshots" tab is the list of active snapshots of the share. This list is divided into two tabs: the "Snapshots" tab is used for browsing and managing snapshots. The "Schedules" tab manages automatic snapshot schedules. Within the "Snapshots" tab, you can select between viewing all snapshots, only manual snapshots, or only scheduled snapshots. For each snapshot, the following fields are shown:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the snapshot. There are two types of snapshots: manual and automatic.</td>
</tr>
<tr>
<td></td>
<td>Manual Snapshots: &quot;Name&quot; is the name provided when the snapshot was created. Manual snapshots can be renamed by clicking on the name and entering a new value.</td>
</tr>
<tr>
<td></td>
<td>Automatic Snapshots: There are three types, and they cannot be renamed:</td>
</tr>
<tr>
<td></td>
<td>- .auto: User-configured scheduled snapshots with custom retention policies (see “Scheduled Snapshots” on page 336).</td>
</tr>
<tr>
<td></td>
<td>- .ndmp: Used for NDMP backup and automatically pruned.</td>
</tr>
<tr>
<td></td>
<td>- .rr: Used for remote replication and automatically pruned.</td>
</tr>
<tr>
<td>Creation</td>
<td>The date and time when the snapshot was created.</td>
</tr>
<tr>
<td>Unique</td>
<td>The amount of unique space used by the snapshot. Snapshots begin initially referencing all the same blocks as the filesystem or LUN itself. As the active filesystem diverges, blocks that have been changed in the active share may remain held by one or more snapshots. When a block is part of multiple snapshots, it will be accounted in the share snapshot usage, but will not appear in the unique space of any particular snapshot. The unique space is blocks that are only held by a particular snapshot, and represents the amount of space that would be freed if the snapshot were to be destroyed.</td>
</tr>
<tr>
<td>Total</td>
<td>The total amount of space referenced by the snapshot. This represents the size of the filesystem at the time the snapshot was taken, and any snapshot can theoretically take up an amount of space equal to the total size as data blocks are rewritten.</td>
</tr>
</tbody>
</table>
Create a Project Level Snapshot using the BUI

1. Open the project that you want to snapshot.
2. Click the Snapshots tab.
3. Click the icon. The snapshots list appears.
4. In the dialog box, type a name for the snapshot.
5. To create the snapshot, click "apply".

Create a Share/LUN Level Snapshot using the BUI

1. Open the share/LUN that you want to snapshot.
2. Click the Snapshots tab.
3. Click the icon. The snapshots list appears.
4. In the dialog box, type a name for the snapshot.
5. To create the snapshot, click "apply".

There is no limit on the number of snapshots that can be taken, but each snapshot consumes memory, so creating large numbers of snapshots can slow the system. The practical limit on the number of snapshots system-wide depends on the system configuration, but should be over a hundred thousand.

Renaming a Snapshot using the BUI

1. To rename a snapshot, click the name within the list of active snapshots. This will change to a text input box.
2. After updating the name within the text input, hitting return or changing focus will commit the changes.

▼ Destroying a Snapshot using the BUI

1. To destroy a snapshot, click the icon when over the row for the target snapshot.

2. Destroying a snapshot will require destroying any clones and their descendents. If this is the case, you will be prompted with a list of the clones that will be affected.

▼ Rolling back to a Snapshot using the BUI

1. To rollback a filesystem, click the icon for the destination snapshot.

2. A confirmation dialog will appear, and if there are any clones of the snapshot, any newer snapshots, or their descendents, they will be displayed, indicating that they will be destroyed as part of this process.

In addition to accessing the data in a filesystem snapshot directory, snapshots can also be used to roll back to a previous instance of the filesystem or LUN. This requires destroying any newer snapshots and their clones, and reverts the share contents to what they were at the time the snapshot was taken. It does not affect any property settings on the share, though changes to filesystem root directory access will be lost, as that is part of the filesystem data.

▼ Cloning a Snapshot using the BUI

- To create a clone, click the icon for the source snapshot. A dialog will prompt for the following values.
  - **Project** - Destination project. By default, clones are created within the current project, but they can also be created in different projects (or moved between projects).
  - **Name** - Type a name for the clone.
  - **Mountpoint** - To use this value, click the lock icon. Set the mountpoint for the clone. When Retain Other Local Settings is set, the clone must be given a different mountpoint, as shares cannot save the same mountpoint.
  - **Resource Name** - To use this value, click the lock icon. Enter the resource that you want to use for the clone.
■ Retain Other Local Settings - By default, all currently inherited properties of the filesystem will inherit from the destination project in the clone. Local settings are always preserved. Setting this property causes any inherited properties to be preserved as local setting in the new clone.

A clone is a writable copy of a snapshot, and is managed like any other share. Like snapshots of filesystems, it initially consumes no additional space. As the data in the clone changes, it will consume more space. The original snapshot cannot be destroyed without also destroying the clone. Scheduled snapshots can be safely cloned, and scheduled snapshots with clones will be ignored if they otherwise should be destroyed.

Scheduling Snapshots using the BUI

In addition to manual snapshots, you can configure automatic snapshots according to the table below. These snapshots are named ".auto-<timestamp>", and can be taken on half hourly, hourly, daily, weekly, or monthly schedules. A schedule is a list of intervals and retention policies. Times are displayed in the local (client browser) time zone. However, times are stored and executed in UTC format and without regard to such conventions as daylight saving time. For example, a snapshot scheduled for 10:00 a.m. PST (UTC-8) is stored and executed at 18:00 UTC.

Automatic snapshots can be set on a project or a share, but not both. Otherwise, overlapping schedules and retention policies would make it impossible to guarantee both schedules. Removing an interval, or changing its retention policy, will immediately destroy any automatic snapshots not covered by the new schedule. Automatic snapshots with clones are ignored.

Previous versions of the software allowed for automatic snapshots at the frequency of a minute. This proved to put undue strain on the system and was not generally useful. To help users avoid placing undue stress on the system, this feature was removed with the 2010.Q3 release. Snapshots can now only be specified at a period of once every half hour or longer. Existing minute periods will be preserved should the software be rolled back, and previous instances will expire according to the existing schedule, but no new snapshots will be taken. An alert will be posted if a share or project with this frequency is found.

To add a new interval, click the icon when viewing the "Schedules" tab. Each interval has the following properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>One of &quot;half hour&quot;, &quot;hour&quot;, &quot;day&quot;, &quot;week&quot;, or &quot;month&quot;. This indicates how often the snapshot is taken.</td>
</tr>
<tr>
<td>Offset</td>
<td>This specifies an offset within the frequency. For example, when selecting an hour frequency, snapshots can be taken at an explicit minute offset from the hour. For daily snapshots, the offset can specify hour and</td>
</tr>
</tbody>
</table>
Manual Snapshots using the CLI

### Property | Description
--- | ---
minute, and for weekly or monthly snapshots the offset can specify day, hour, and minute. |  
Keep at most | Controls the retention policy for snapshots. Automatic snapshots can be kept forever (except for half hour and hour snapshots, which are capped at 48 and 24, respectively) or can be limited to a certain number. This limit will delete automatic snapshots for the given interval if they are older than the retention policy. This is actually enforced by the time they were taken, not an absolute count. So if you have hour snapshots and the appliance is down for a day, when you come back up all your hour snapshots will be deleted. Snapshots that are part of multiple intervals are only destroyed when no interval specifies that they should be retained.  

### Manual Snapshots using the CLI

- **To access share snapshots, navigate to the share and the snapshots context.**

  ```
  clownfish: SHARES select default select builds
  clownfish:shares default/builds> snapshots
  clownfish:shares default/builds snapshots>
  ```

### Listing Snapshots using the CLI

- **Snapshots can be listed using the standard CLI commands.**

  ```
  clownfish:shares default/builds snapshots> list
  today
  yesterday
  clownfish:shares default/builds snapshots>
  ```

### Taking Manual Snapshots using the CLI

1. **To take a manual project-level snapshot, navigate to the project and snapshot node and then use the snapshot command.**

   ```
   clownfish:cd /
   clownfish:shares select myproject snapshots
   clownfish:shares myproject snapshots> snapshot cob_monday
   ```
Renaming a Snapshot using the CLI

2. To take a manual share-level snapshot of an individual share, navigate to that share and use the snapshot command there:

```bash
clownfish:cd /
clownfish:shares select myproject select share1 snapshots
clownfish: snapshot lunchtime
```

▼ Renaming a Snapshot using the CLI

● To rename a snapshot, use the `rename` command.

```bash
clownfish:shares default/builds snapshots> rename test test2
clownfish:shares default/builds snapshots>
```

▼ Destroying a Snapshot using the CLI

1. To destroy a snapshot, use the `destroy` command:

```bash
clownfish:shares default/builds snapshots> select test2
clownfish:shares default/builds@test2> destroy
This will destroy this snapshot. Are you sure? (Y/N)
clownfish:shares default/builds snapshots>
```

2. You can also use the `destroy` command from the share context without selecting an individual snapshot:

```bash
clownfish:shares default/builds snapshots> destroy test2
This will destroy this snapshot. Are you sure? (Y/N)
clownfish:shares default/builds snapshots>
```

▼ Rolling back to a Snapshot using the CLI

● To rollback to a snapshot, select the target snapshot and run the `rollback` command:

```bash
clownfish:shares default/builds snapshots> select today
clownfish:shares default/builds@today> rollback
Rolling back will revert data to snapshot, destroying newer data. Active initiators will be disconnected.
```
Cloning a Snapshot using the CLI

1. To clone a snapshot, use the `clone` command. This command will place you into an uncommitted share context identical to the one used to create shares. From here, you can adjust properties as needed before committing the changes to create the clone.

    clownfish:shares default/builds snapshots> select today
    clownfish:shares default/builds@today> clone testbed
    clownfish:shares default/testbed (uncommitted clone)> get
        aclinherit = restricted (inherited)
        aclmode = discard (inherited)
        atime = true (inherited)
        checksum = fletcher4 (inherited)
        compression = off (inherited)
        copies = 1 (inherited)
        mountpoint = /export/testbed (inherited)
        quota = 0 (default)
        readonly = false (inherited)
        recordsize = 128K (inherited)
        reservation = 0 (default)
        secondarycache = all (inherited)
        nbmand = false (inherited)
        sharesmb = off (inherited)
        sharenfs = on (inherited)
        snapdir = hidden (inherited)
        vscan = false (inherited)
        sharedav = off (inherited)
        shareftp = off (inherited)
        root_group = other (default)
        root.permissions = 777 (default)
        root.user = nobody (default)
        quota_snap = true (default)
        reservation_snap = true (default)
    clownfish:shares default/testbed (uncommitted clone)> set quota=10G
        quota = 10G (uncommitted)
    clownfish:shares default/testbed (uncommitted clone)> commit
    clownfish:shares default/builds@today>

2. The command also supports an optional first argument, which is the project in which to create the clone. By default, the clone is created in the same project as the share being cloned.
Listing Dependent Clones using the CLI

- To list all clones created from a particular snapshot (dependent clones), navigate to the snapshot and then use the list clones command. The result shows clone names and the project where the clone resides.

```
clonefish:shares default/builds> snapshots
clonefish:shares default/builds snapshots> select today
clonefish:shares default/builds@today> list clones

Clones: 2 total

<table>
<thead>
<tr>
<th>PROJECT</th>
<th>SHARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>testbed</td>
</tr>
<tr>
<td>default</td>
<td>production</td>
</tr>
</tbody>
</table>
```

Scheduled Snapshots using the CLI

Automatic scheduled snapshots can be configured using the automatic command from the snapshot context, at the project level of that of an individual share. Once in this context, new intervals can be added and removed with the create and destroy commands. Each interval has a set of properties that map to the BUI view of the frequency, offset, and number of snapshots to keep. Schedules are maintained in UTC format.

- To configure automatic scheduled snapshots use the automatic command from the snapshot context, at the project level of that of an individual share.

```
clownfish:shares default/builds snapshots> automatic
clownfish:shares default/builds snapshots automatic> create
clownfish:shares default/builds snapshots automatic (uncommitted)> set frequency=day
  frequency = day (uncommitted)
clownfish:shares default/builds snapshots automatic (uncommitted)> set hour=14
  hour = 14 (uncommitted)
clownfish:shares default/builds snapshots automatic (uncommitted)> set minute=30
  minute = 30 (uncommitted)
clownfish:shares default/builds snapshots automatic (uncommitted)> set keep=7
  keep = 7 (uncommitted)
clownfish:shares default/builds snapshots automatic (uncommitted)> get
  frequency = day (uncommitted)
  day = (unset)
  hour = 14 (uncommitted)
  minute = 30 (uncommitted)
  keep = 7 (uncommitted)
clownfish:shares default/builds snapshots automatic (uncommitted)> commit
```

```
clownfish:shares default/builds snapshots automatic> list

NAME          FREQUENCY          DAY        HH:MM KEEP
```
Setting a Scheduled Snapshot Label using the CLI

In the BUI, the "scheduled snapshot label" property can be set for either a project or a share. Likewise, in the CLI, the label can be set by first navigating to either the project or share context.

To create a scheduled snapshot label, use the `set snaplabel` command:

```
clownfish:shares project1/share1> set snaplabel=project1:share1
```

Working with Replication

**LICENSE NOTICE:** Remote Replication and Cloning may be evaluated free of charge, but each feature requires that an independent license be purchased separately for use in production. After the evaluation period, these features must either be licensed or deactivated. Oracle reserves the right to audit for licensing compliance at any time. For details, refer to the "Oracle Software License Agreement ("SLA") and Entitlement for Hardware Systems with Integrated Software Options."

Oracle ZFS Storage Appliances support snapshot-based replication of projects and shares from a source appliance to any number of target appliances manually, on a schedule, or continuously. The replication includes both data and metadata. Remote replication (or just "replication") is a general-purpose feature optimized for the following use cases:

- **Disaster recovery** - Replication can be used to mirror an appliance for disaster recovery. In the event of a disaster that impacts service of the primary appliance (or even an entire data center), administrators activate service at the disaster recovery site, which takes over using the most recently replicated data. When the primary site has been restored, data changed while the disaster recovery site was in service can be migrated back to the primary site and normal service restored. Such scenarios are fully testable before such a disaster occurs.

- **Data distribution** - Replication can be used to distribute data (such as virtual machine images or media) to remote systems across the world in situations where clients of the target appliance wouldn't ordinarily be able to reach the source appliance directly, or such a setup would have prohibitively high latency. One example uses this scheme for local caching to improve latency of read-only data (like documents).

- **Disk-to-disk backup** - Replication can be used as a backup solution for environments in which tape backups are not feasible. Tape backup might not be feasible, for example,
because the available bandwidth is insufficient or because the latency for recovery is too high.

- **Data migration** - Replication can be used to migrate data and configuration between appliances when upgrading hardware or rebalancing storage. Shadow migration can also be used for this purpose.

To configure replication, use the following sections:

- “Understanding Replication” on page 339
- “Understanding Project Replication Targets” on page 340
- “Understanding Project Replication Actions and Packages” on page 341
- “Understanding Project Replication Storage Pools” on page 342
- “Understanding Project vs. Share Replication” on page 343
- “Understanding Replication Configuration Details” on page 343
- “Replication Authorizations” on page 344
- “Replication Alerts and Audit Events” on page 345
- “Understanding Replication and Clustering” on page 345
- “Understanding Replication Snapshots and Data Consistency” on page 346
- “Managing Replication Snapshots” on page 346
- “Understanding iSCSI Configurations and Replication” on page 348
- “Replicating Clones” on page 348
- “Using Replication Analytics” on page 348
- “Understanding Replication Failures” on page 349
- “Understanding Appliance Upgrades and Replication” on page 351
- “Creating and Editing Replication Actions” on page 352
- “Creating and Editing Replication Targets using the BUI” on page 354
- “Creating and Editing Replication Targets using the CLI” on page 355
- “Creating and Editing Replication Actions using the BUI” on page 356
- “Creating and Editing Replication Actions using the CLI” on page 356
- “Managing Replication Packages” on page 358
- “Managing Replication Packages using the BUI” on page 359
- “Managing Replication Packages using the CLI” on page 360
- “Canceling Replication Updates” on page 361
- “Disabling a Replication Package” on page 362
- “Cloning a Replication Package or Share” on page 362
- “Exporting Replicated Filesystems” on page 363
- “Severing Replication” on page 364
- “Reversing the Direction of Replication” on page 365
- “Destroying a Replication Package” on page 366
- “Reversing Replication using the BUI” on page 366
- “Reversing Replication for Disaster Recovery using the BUI” on page 368
Remote replication has the following important properties:

- **Snapshot-based** - The replication subsystem takes a snapshot as part of each update operation. For a full update, the entire project contents up to the snapshot are sent. For an incremental update, only the changes since the last replication snapshot for the same action are sent.

- **Block-level** - Each update operation traverses the filesystem at the block level and sends the appropriate filesystem data and metadata to the target.

- **Asynchronous** - Because replication takes snapshots and then sends them, data is necessarily committed to stable storage before replication even begins sending it. Continuous replication effectively sends continuous streams of filesystem changes, but it's still asynchronous with respect to NAS and SAN clients.

- **Includes metadata** - The underlying replication stream serializes both user data and ZFS metadata, including most properties configured on the Shares screen. These properties can be modified on the target after the first replication update completes, though not all take effect until the replication connection is severed. For example, this allows sharing over NFS to a different set of hosts than on the source. See “Managing Replication Packages” on page 358 for details.

- **Secure** - The replication control protocol used among ZFS Storage Appliances is secured with SSL. Data can optionally be protected with SSL as well. Appliances can only replicate to/from other appliances after an initial manual authentication process, see “Creating and Editing Replication Targets using the CLI” on page 355.

Replication has the following known limitations:

- Changing a target's IP address will break the replication
- Actions cannot move between pools
- I/O is limited to a maximum of 200 MB/s per project level replication

The following is a list of the common replication terms.

- **Replication Peer** (or just peer, in this context) - a ZFS Storage Appliance that has been configured as a replication source or target.

- **Replication Source** (or just source) - an appliance peer containing data to be replicated to another appliance peer (the target). Individual appliances can act as both a source and a target, but are only one of these in the context of a particular replication action.

- **Replication Target** (or just target) - an appliance peer that will receive and store data replicated from another appliance peer (the source). This term also refers to a configuration object on the appliance that enables it to replicate to another appliance.
- **Replication Group** (or just group) - the set of datasets (exactly one project and some number of shares) which are replicated as a unit. See “Understanding Project vs. Share Replication” on page 343.

- **Replication Action** (or just action) - a configuration object on a source appliance specifying a project or share, a target appliance, and policy options (including how often to send updates, whether to encrypt data on the wire, etc.).

- **Package** - the target-side analog of an action; the configuration object on the target appliance that manages the data replicated as part of a particular action from a particular source. Each action on a source appliance is associated with exactly one package on a target appliance and vice versa. Loss of either object will require creating a new action/package pair (and a full replication update).

- **Full Sync** (or full update) - a replication operation that sends the entire contents of a project and some of its shares.

- **Incremental Update** - a replication operation that sends only the differences in a project and its shares since the previous update (whether that one was full or incremental).

### Understanding Project Replication Targets

Before a source appliance can replicate to a target, the two systems must set up a replication peer connection that enables the appliances to identify each other securely for future communications. Administrators set up this connection by creating a new replication target on the Configuration > Services > Remote Replication screen on the source appliance. To create a new target, administrators specify three fields:

- Name (used only to identify the target in the source appliance's BUI and CLI)
- Network address or hostname (to contact the target appliance)
- Target appliance's root password (to authorize the administrator to set up the connection on the target appliance)

The appliances then exchange keys used to securely identify each other in subsequent communications. These keys are stored persistently as part of the appliance's configuration and persist across reboots and upgrades. They will be lost if the appliance is factory reset or reinstalled. The root password is never stored persistently, so changing the root password on either appliance does not require any changes to the replication configuration. The password is never transmitted in the clear because this initial identity exchange (like all replication control operations) is protected with SSL.

By default, the replication target connection is not bidirectional. If an administrator configures replication from a source A to a target B, B cannot automatically use A as a target. However, the system supports reversing the direction of replication, which automatically creates a target for A on B (if it does not already exist) so that B can replicate back to A.

**NOTE:** When a replication source uses NIS or LDAP services to map users or user groups and those users or user groups are included in a share configuration on the source (for example
in ‘Share Level ACL’ or ‘Share Space Usage’), those users or user groups should be available on the replication target (for example by using the same NIS or LDAP servers); otherwise replication sever/reverse operations may fail.

To configure replication targets, see “Cloning a Received Replication Project using the CLI” on page 373.

**Understanding Project Replication Actions and Packages**

Targets represent a connection between appliances that enables them to communicate securely for the purpose of replication, but targets do not specify what will be replicated, how often, or with what options. For this, administrators must define replication actions on the source appliance. Actions are the primary administrative control point for replication, each one specifying:

- A replication group (a project and some number of shares)
- A target appliance
- A storage pool on the target appliance (used only during the initial setup)
- A frequency (which may be manual, scheduled, or continuous)
- Additional options, such as whether to encrypt the data stream on the wire

The group is specified implicitly by the project or share on which the action is configured (see “Understanding Project Replication Storage Pools” on page 342). The target appliance and storage pool cannot be changed after the action is created, but the other options can be modified at any time. Generally, if a replication update is in progress when an option is changed, then the new value only takes effect when the next update begins.

Actions are the primary unit of replication configuration on the appliance. Each action corresponds to a package on the target appliance that contains an exact copy of the source project and shares on which the action is configured as of the start time of the last replication update. Administrators configure the frequency and other options for replication updates by modifying properties of the corresponding action. Creating the action on the source appliance creates the package on the target appliance in the specified storage pool, so the source must be able to contact the target when the action is initially created.

The first update for each replication action sends a full sync (or full update): the entire contents of the action’s project and shares are sent to the target appliance. Once this initial sync completes, subsequent replication updates are incremental: only the changes since the previous update are sent. The action (on the source) and package (on the target) keep track of which changes have been replicated to the target through named replication snapshots. Generally, as long as at least one full sync has been sent for an action and the action/package connection has not been corrupted due to a software failure or administrative action, replication updates will be incremental.
The action and package are bound to each other. If the package is somehow corrupted or destroyed, the action will not be able to send replication updates, even if the target still has the data and snapshots associated with the action. Similarly, if the action is destroyed, the package will be unable to receive new replication updates (even if the source still has the same data and snapshots). The BUI and CLI warn administrators attempting to perform operations that would destroy the action-package connection. If an error or explicit administrative operation breaks the action-package connection such that an incremental update is no longer possible, administrators must sever or destroy the package and action and create a new action on the source.

**Note** - The appliance avoids destroying data on the target unless explicitly requested by the administrator. As a result, if the initial replication update for an action fails after replicating some of the data and leaving incomplete data inside the package, subsequent replication updates using the same action will fail because the appliance cannot overwrite the previously received data. To resolve this, administrators should destroy the existing action and package and create a new action and package and start replication again.

In software releases prior to 2010.Q1, action and replica configuration (like target configuration) was stored on the controller rather than as part of the project and share configuration in the storage pool. As a result, a factory reset caused configuration to be destroyed. In 2010.Q1 and later releases, the action and package configuration is stored in the storage pool with the corresponding projects and shares and will be available even after a factory reset. However, target information will still be lost, and actions with missing targets currently cannot be configured to point to a new target.

**Understanding Project Replication Storage Pools**

When the action is initially configured, the administrator is given a choice of which storage pool on the target should contain the replicated data. The storage pool containing an action cannot be changed once the action has been created. Creating the action creates the empty package on the target in the specified storage pool, and after this operation the source has no knowledge of the storage configuration on the target. It does not keep track of which pool the action is being replicated to, nor is it updated with storage configuration changes on the target.

When the target is a clustered system, the chosen storage pool must be one owned by same head which owns the IP address used by the source for replication because only those pools are always guaranteed to be accessible when the source contacts the target using that IP address. This is exactly analogous to the configuration of NAS clients (NFS and SMB), where the IP address and path requested in a mount operation must obey the same constraint. When performing operations that change the ownership of storage pools and IP addresses in a cluster, administrators must consider the impact to sources replicating to the cluster. There is currently no way to move packages between storage pools.
Understanding Project vs. Share Replication

The appliance allows administrators to configure remote replication on both the project and share level. Like other properties configurable on the Shares screen, each share can either inherit or override the configuration of its parent project. Inheriting the configuration means not only that the share is replicated on the same schedule to the same target with the same options as its parent project is, but also that the share will be replicated in the same stream using the same project-level snapshots as other shares inheriting the project's configuration. This may be important for applications which require consistency between data stored on multiple shares. Overriding the configuration means that the share will not be replicated with any project-level actions, though it may be replicated with its own share-level actions that will include the project. It is not possible to override part of the project's replication configuration and inherit the rest.

More precisely, the replication configuration of a project and its shares define some number of replication groups, each of which is replicated with a single stream using snapshots taken simultaneously. All groups contain the project itself (which essentially just includes its properties). One project-level group includes all shares inheriting the replication configuration of the parent project. Any share that overrides the project's configuration forms a new group consisting of only the project and the share itself.

For example, suppose we have a project home and shares bill, cindi, and dave, where:

- home has replication configured with some number of actions
- home/bill and home/cindi inherit the project's replication configuration
- home/dave overrides the project's replication configuration, using its own configuration with some number of actions

This configuration defines the following replication groups, each of which is replicated as a single stream per action using snapshots taken simultaneously on the project and shares:

- One project-level group including home, home/bill, and home/cindi.
- One share-level group including home and home/dave.

Due to current limitations, do not mix project- and share-level replications within the same project. This avoids unpredictable results when reversing the replication direction or when replicating clones. For more details, see “Managing Replication Packages” on page 358 and “Replicating Clones” on page 348.

Understanding Replication Configuration Details

- **Scheduled vs. Continuous Replication** - Replication actions can be configured to send updates on a schedule or continuously. The replication update process itself is the same in both cases. This property only controls the interval.
Because continuous replication actions send updates as frequently as possible, they result in sending a constant stream of all filesystem changes to the target system. For filesystems with a lot of churn (many files created and destroyed in short intervals), this can result in replicating much more data than actually necessary. However, as long as replication can keep up with data changes, this results in the minimum data lost in the event of a data-loss disaster on the source system.

Continuous replication is still asynchronous. ZFS Storage Appliances do not currently support synchronous replication, which does not consider data committed to stable storage until it’s committed to stable storage on both the primary and secondary storage systems.

- **Intermediate Replication Snapshots** - When the "Include Snapshots" property is true, replication updates include the non-replication snapshots created after the previous replication update (or since the share's creation, in the case of the first full update). This includes automatic snapshots and administrator-created snapshots. This property can be disabled to skip these snapshots and send only the changes between replication snapshots with each update.

- **Sending and Canceling Replication Updates** - For targets that are configured with scheduled or manual replication, administrators can immediately send a replication update by clicking the button in the BUI or using the `sendupdate` command in the CLI. This is not available (or will not work) if an update is actively being sent. Ensure there is enough disk space on the target to replicate the entire project before sending an update. If an update is currently active, the BUI displays a progress bar, and the CLI shows a state of sending. To cancel the update, click the button or use the `cancelupdate` command. It may take several seconds before the cancellation completes.

**Replication Authorizations**

In addition to the Remote Replication filter under the Services scope that allows administrators to stop, start, and restart the replication service, the replication subsystem provides two user authorizations under the "Projects and Shares" scope:

<table>
<thead>
<tr>
<th>Authorization</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>rresource</td>
<td>Allows administrators to create, edit, and destroy replication targets and actions and send and cancel updates for replication actions.</td>
</tr>
<tr>
<td>rrtarget</td>
<td>Allows administrators to manage replicated packages, including disabling replication at the package level, cloning a package or its members, modifying properties of received datasets, and severing or reversing replication. Other authorizations may be required for some of these operations (like setting properties or cloning individual shares). See the available authorizations in the Projects and Shares scope for details.</td>
</tr>
</tbody>
</table>
The `rrsource` authorization is required to configure replication targets on an appliance, even though this is configured under the Remote Replication service screen. For help with authorizations, see “Configuring Users” on page 124.

### Replication Alerts and Audit Events

The system posts alerts when any of the following events occur:

- Manual or scheduled replication update starts or finishes successfully (both source and target).
- Any replication update fails, including as a result of explicit cancellation by an administrator (both source and target).
- A scheduled replication update is skipped because another update for the same action is already in progress (see above).
- When a Continuous replication starts for the first time.
- When a Continuous replication fails.
- When a continuous replication starts for the first time, fails, or resumes after a failure.

The system audits the following replication events and records them in the logs.

- Creating, modifying or destroying replication actions
- Adding or removing shares from a replication group
- Creating, modifying, cloning, reversing, severing or destroying replication packages on the target
- Creating, modifying or destroying replication targets

### Understanding Replication and Clustering

Replication can be configured from any ZFS Storage Appliance to any other ZFS Storage Appliance regardless of whether each is part of a cluster and whether the appliance's cluster peer has replication configured in either direction, except for the following constraints:

- Configuring replication from both peers of a cluster to the same replication target is unsupported, but a similar configuration can be achieved using two different IP addresses for the same target appliance. Administrators can use the multiple IP addresses of the target appliance to create one replication target on each cluster head for use by that head.
- When configuring replication between cluster peers, configure replication with both controllers in the CLUSTERED state. Do not use private network addresses and use separate replication targets for each controller's pools.

The following rules govern the behavior of replication in clustered configurations:

- Replication updates for projects and shares are sent from whichever cluster peer has imported the containing storage pool.
■ Replication updates are received by whichever peer has imported the IP address configured in the replication action on the source. Administrators must ensure that the head using this IP address will always have the storage pool containing the replica imported. This is ensured by assigning the pool and IP address resources to the same head during cluster configuration.

■ Replication updates (both to and from an appliance) that are in progress when an appliance exports the corresponding storage pool or IP address (as part of a takeover or failback) will fail. Replication updates using storage pools and IP addresses unaffected by a takeover or failback operation will be unaffected by the operation.

For details on clustering and cluster terminology, review “Configuring Clustering” on page 141.

Understanding Replication Snapshots and Data Consistency

The appliance replicates snapshots and each snapshot is received atomically on the target, so the contents of a share's replica on the target always matches the share's contents on the source at the time the snapshot was taken. Because the snapshots for all shares sent in a particular group are taken at the same time (see above), the entire package contents after the completion of a successful replication update exactly matches the group's content when the snapshot was created on the source (when the replication update began).

However, each share's snapshots are replicated separately (and serially), so it's possible for some shares within a package to have been updated with a snapshot that is more recent than those of other shares in the same package. This is true during a replication update (after some shares have been updated but before others have) and after a failed replication update (after which some shares may have been updated but others may not have been).

To summarize:

■ Each share is always point-in-time consistent on the target (self-consistent).

■ When no replication update is in progress and the previous replication update succeeded, each package's shares are also point-in-time consistent with each other (package-consistent).

■ When a replication update is in progress or the previous update failed, package shares may be inconsistent with each other, but each one will still be self-consistent. If package consistency is important for an application, one must clone the replication package, which always clones the most recent successfully received snapshot of each share.

Managing Replication Snapshots

Snapshots are the basis for incremental replication. The source and target must always share a common snapshot in order to continue replicating incrementally, and the source must know
which is the most recent snapshot that the target has. To facilitate this, the replication subsystem
creates and manages its own snapshots. Administrators generally need not be concerned with
them, but the details are described here since snapshots can have significant effects on storage
utilization.

Each replication update for a particular action consists of the following steps:

- Determine whether this is an incremental or full update based on whether we've tried to
  replicate this action before and whether the target already has the necessary snapshot for an
  incremental update.
- Take a new project-level snapshot.
- Send the update. For a full update, send the entire group's contents up to the new snapshot.
  For an incremental update, send the difference between from the previous (base) snapshot
  and the new snapshot.
- Record the new snapshot as the base snapshot for the next update and destroy the previous
  base snapshot (for incremental updates). The base snapshot remains on the target until the
  next update is received at which point it is the first thing that is destroyed.

This has several consequences for snapshot management:

- During the first replication update and after the initial update when replication is not
  active, there is exactly one project-level snapshot for each action configured on the project
  or any share in the group. A replication action may create snapshots on shares that are in
  the same project as the share(s) in the group being replicated by the action but that are not
  being sent as part of the update for the group.
- During subsequent replication updates of a particular action, there may be two project-
  level snapshots associated with the action. Both snapshots may remain after the update
  completes in the event of failure where the source was unable to determine whether the
  target successfully received the new snapshot (as in the case of a network outage during
  the update that causes a failure).
- None of the snapshots associated with a replication action can be destroyed by the
  administrator without breaking incremental replication. The system will not allow
  administrators to destroy snapshots on either the source or target that are necessary for
  incremental replication. To destroy such snapshots on the source, one must destroy the
  action (which destroys the snapshots associated with the action). To destroy such snapshots
  on the target, one must first sever the package (which destroys the ability to receive
  incremental updates to that package).
- Administrators must not rollback to snapshots created prior to any replication snapshots.
  Doing so will destroy the later replication snapshots and break incremental replication for
  any actions using those snapshots.
- Replication's usage of snapshots requires that administrators using replication understand
  space management on the appliance, particularly as it applies to snapshots. See “Space
  Management for Shares” on page 287.
- For information about space management for replicating LUNs, see “Space Management
  for Replicating LUNs” on page 287.
Understanding iSCSI Configurations and Replication

Replication updates include most of the configuration specified on the Shares screen for a project and its shares. This includes any target groups and initiator groups associated with replicated LUNs. When using non-default target groups and initiator groups, administrators must ensure that the target groups and initiator groups used by LUNs within the project also exist on the replication target. It is only required that groups exist with the same name, not that they define the same configuration. Failure to ensure this can result in failure to clone and export replicated LUNs.

The SCSI GUID associated with a LUN is replicated with the LUN. As a result, the LUN on the target appliance will have the same SCSI GUID as the LUN on the source appliance. Clones of replicated LUNs, however, will have different GUIDs (just as clones of local LUNs have different GUIDs than their origins).

Replicating Clones

Replication in 2009.Q3 and earlier was project-level only and explicitly disallowed replicating projects containing clones whose origin snapshots resided outside the project. With share-level replication in 2010.Q1 and later, this restriction has been relaxed, but administrators must still consider the origin snapshots of clones being replicated. In particular, the initial replication of a clone requires that the origin snapshot have already been replicated to the target or is being replicated as part of the same update. This restriction is not enforced by the appliance management software, but attempting to replicate a clone when the origin snapshot does not exist on the target will fail.

In practice, there are several ways to ensure that replication of a clone will succeed:

- If the clone's origin snapshot is in the same project, just use project-level replication.
- If the clone's origin snapshot is not in the same project or if project-level replication that includes the origin is undesirable for other reasons, use share-level replication to replicate the origin share first and then use project-level or share-level replication to replicate the clone.
- Do not destroy the clone's origin on the target system unless you intend to also destroy the clone itself.

In all cases, the “include snapshots” property should be true on the origin's action to ensure that the origin snapshot is actually sent to the target.

Using Replication Analytics

The following Analytics are available for replication:
For more information about Analytics, see the “Oracle ZFS Storage Appliance Analytics Guide”.

### Understanding Replication Failures

Individual replication updates can fail for a number of reasons. Where possible, the appliance reports the reason for the failure in alerts posted on the source appliance or target appliance, or on the Replication screen for the action that failed. You may be able to get details on the failure by clicking the orange alert icon representing the action's status. The following are the most common types of failures:

<table>
<thead>
<tr>
<th>Failure</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancelled</td>
<td>The replication update was cancelled by an administrator. Replication can be cancelled on the source or target and it's possible for one peer not to realize that the other peer has cancelled the operation.</td>
</tr>
<tr>
<td>Network connectivity failure</td>
<td>The appliance was unable to connect to the target appliance due to a network problem. There may be a misconfiguration on the source, target, or the network.</td>
</tr>
<tr>
<td>Peer verification failed</td>
<td>The appliance failed to verify the identity of the target. This occurs most commonly when the target has been reinstalled or factory reset. A new replication target must be configured on the source appliance for a target which has been reinstalled or factory reset in order to generate a new set of authentication keys. See “Managing Replication Packages” on page 350.</td>
</tr>
<tr>
<td>Peer RPC failed</td>
<td>A remote procedure call failed on the target system. This occurs most commonly when the target appliance is running incompatible software. For details, see “Understanding Appliance Upgrades and Replication” on page 351.</td>
</tr>
<tr>
<td>No package</td>
<td>Replication failed because no package exists on the target to contain the replicated data. Since the package is created when configuring the action, this error typically happens after an administrator has destroyed the package on the target. It's also possible to see this error if the storage pool containing the package is not imported on the target system, which may occur if the pool is faulted or if storage or networking has been reconfigured on the target appliance.</td>
</tr>
<tr>
<td>Non-empty package exists</td>
<td>Replication failed because the target package contains data from a previous, failed replication update. This error occurs when attempting to send a replication update for an action whose first replication update failed after</td>
</tr>
</tbody>
</table>
### Setting a Scheduled Snapshot Label using the CLI

<table>
<thead>
<tr>
<th>Failure</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replicating some data</td>
<td>The target appliance will not destroy data without explicit administrative direction, so it will not overwrite the partially received data. The administrator should remove the existing action and package and create a new action on the source and start replication again.</td>
</tr>
<tr>
<td>Disabled</td>
<td>Replication failed because it is disabled on the target. Either the replication service is disabled on the target or replication has been disabled for the specific package being replicated.</td>
</tr>
<tr>
<td>Target busy</td>
<td>Replication failed because the target system has reached the maximum number of concurrent replication updates. The system limits the maximum number of ongoing replication operations to avoid resource exhaustion. When this limit is reached, subsequent attempts to receive updates will fail with this error, while subsequent attempts to send updates will queue up until resources are available.</td>
</tr>
<tr>
<td>Out of space</td>
<td>Replication failed because the source system had insufficient space to create a new snapshot. This may be because there is no physical space available in the storage pool or because the project or one of its shares would be over quota because of reservations that don't include snapshots.</td>
</tr>
<tr>
<td>Key Unavailability</td>
<td>Replication failed because the encryption key used by the share is not available either on the source or target system. Review the alerts on both the source and target systems to ensure the key is available on both systems. See “Replicating an Encrypted Share” on page 413 for information about replicating encrypted shares and projects.</td>
</tr>
<tr>
<td>Incompatible target</td>
<td>Replication failed because the target system is unable to receive the source system's data stream format. This can happen as a result of upgrading a source system and applying deferred updates without having upgraded and applied the same updates on the target. Check the release notes for the source system's software version for a list of deferred updates and whether any have implications for remote replication.</td>
</tr>
<tr>
<td>Misc</td>
<td>Replication failed, but no additional information is available on the source. Check the alert log on the target system and if necessary contact support for assistance. Some failure modes that currently fall into this category include insufficient disk space on the target to receive the update and attempting to replicate a clone whose origin snapshot does not exist on the target system.</td>
</tr>
</tbody>
</table>

A replication update fails if any part of the update fails. The current implementation replicates the shares inside a project serially and does not rollback changes from failed updates. As a result, when an update fails, some shares on the target may be up-to-date while others are not. See "Snapshots and Data Consistency" above for details.
Although some data may have been successfully replicated as part of a failed update, the current implementation resends all data that was sent as part of the previous (failed) update. That is, failed updates will not pick up where they left off, but rather will start where the failed update started.

When scheduled or continuous replication fails, the system waits several minutes and tries again. The system will continue retrying failed scheduled or continuous replications indefinitely. At any point during the retry procedure, initiating a manual update will immediately begin a retry, circumventing the usual delay between successive retries. If the manual update completes successfully, it terminates the retry sequence and the replication action reverts to its normal scheduled or continuous updates. Failed manual replications without a prior scheduled replication action are not retried.

When a replication update is in progress and another update is scheduled, the scheduled replication is deferred until the previous update completes, and an alert is posted.

**Understanding Appliance Upgrades and Replication**

The replication implementation has changed significantly between the 2009.Q3 and 2010.Q1 releases. It remains highly recommended to suspend replication to and from an appliance before initiating an upgrade from 2009.Q3 or earlier. This is mandatory in clusters using rolling upgrade.

There are three important user-visible changes related to upgrade to 2010.Q1 or later:

- The network protocol used for replication has been enhanced. 2009.Q3 systems can replicate to systems running any release (including 2010.Q1 and later), while systems running 2010.Q1 or later can only replicate to other systems running 2010.Q1 or later. In practice, this means that replication targets must be upgraded before or at the same time as their replication sources to avoid failures resulting from incompatible protocol versions.

- Replication action configuration is now stored in the storage pool itself rather than on the head system. As a result, after upgrading from 2009.Q3 or earlier to 2010.Q1, administrators must apply the deferred updates to migrate their replication configuration.

- Until these updates are applied, incoming replication updates for existing replicas will fail, and replication updates will not be sent for actions configured under 2009.Q3 or earlier. Additionally, space will be used in the storage pool for unmigrated replicas that are not manageable from the BUI or CLI.

- Once these updates are applied, as with all deferred updates, rolling back the system software will have undefined results. It should be expected that under the older release, replicated data will be inaccessible, all replication actions will be unconfigured, and incoming replication updates will be full updates.

- Replication authorizations have been moved from their own scope into the Projects and Shares scope. Any replication authorizations configured on 2009.Q3 or earlier
Setting a Scheduled Snapshot Label using the CLI

will no longer exist under 2010.Q1. Administrators using fine-grained access control for replication should delegate the new replication authorizations to the appropriate administrators after upgrading.

Before performing a replication update, the replication service verifies that the target system is compatible with new data from the source.

- If there are features in use on the source that are not compatible with the target and the features can be safely disabled, the replication service will disable the features, perform the update, and issue a warning.
- If there are features in use on the source that are not compatible with the target and cannot be disabled, the replication service will not perform the update and issue an error.

**Note** - It is always best to upgrade the target as soon as possible.

Updates that break replication compatibility, are delivered as Deferred Updates. For the current list and description, see “Updates” in "Oracle ZFS Storage Appliance Customer Service Manual" and the Oracle ZFS Storage Appliance Release Notes for your current release.

**Creating and Editing Replication Actions**

Replication actions have the following properties, which are presented slightly differently in the BUI and CLI:
FIGURE 44  Add Replication Action

![Add Replication Action](image)

TABLE 139  Replication Action CLI Properties

<table>
<thead>
<tr>
<th>Property (CLI name)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>Unique identifier for the replication target system. This property is specified when an action is initially configured and immutable thereafter.</td>
</tr>
<tr>
<td>Pool</td>
<td>Storage pool on the target where this project will be replicated. This property is specified when an action is initially configured and not shown thereafter.</td>
</tr>
<tr>
<td>Mode (CLI: continuous) and schedule</td>
<td>Whether this action is being replicated continuously or at manual or scheduled intervals. See “Understanding Project Replication Actions and Packages” on page 341 for details.</td>
</tr>
<tr>
<td>Include Snapshots</td>
<td>Whether replication updates include non-replication snapshots. See “Understanding Project Replication Actions and Packages” on page 341 for details.</td>
</tr>
<tr>
<td>Limit bandwidth</td>
<td>Specifies a maximum speed for this replication update (in terms of amount of data transferred over the network).</td>
</tr>
</tbody>
</table>
Creating and Editing Replication Targets using the BUI

1. To create remote replication targets in the BUI, go to Configuration > Services > Remote Replication > Targets. Click Targets and configure a Name, Hostname and Password.

2. To edit remote replication targets in the BUI, go to Configuration > Services > Remote Replication > Targets. For the target you want to edit, move the cursor over the target name, click the pencil icon, and configure the Name and/or Hostname. The Hostname must resolve to the same appliance as before (checked by the serial number of the target). If you want to point to a different appliance than previously configured, you must create a new target to authenticate against the new appliance.
Creating and Editing Replication Targets using the CLI

1. In the CLI, navigate to the targets node to set or unset the target hostname, root_password, and label.
   
   knife:> configuration services replication targets

2. From this context, administrators can:
   ■ Add new targets
   ■ View the actions configured with the existing target
   ■ Edit the unique identifier (label) and/or hostname for the target
   ■ Destroy a target, if no actions are using it

3. A target should not be destroyed while actions are using it. Such actions will be permanently broken. The system makes a best effort to enforce this but cannot guarantee that no actions exist in exported storage pools that are using a given target.

   Note - To replicate an encrypted share, the target must support encryption. See “Replicating an Encrypted Share” on page 413.
Creating and Editing Replication Actions using the BUI

1. After at least one replication target has been configured, administrators can configure actions on a local project or share by navigating to it in the BUI and clicking the Replication tab or navigating to it in the CLI and selecting the "replication" node. These interfaces show the status of existing actions configured on the project or share, the replication progress information, and allow administrators to create new actions.

2. When replicating to a target, two rows of status information are displayed. The first row shows the target name, the date and time of the last successful synchronization, and a progress bar, or a barber-pole progress bar if the replication is continuous. The second row displays the replication type (Scheduled, Manual, or Continuous), the date and time of the last attempted or failed synchronization, and status details. For replications in progress, the status details contain the percentage of completion, the estimated size of the data to be replicated, the average replication throughput, and the estimated completion time. When replication is not in progress, the Status column displays either the next scheduled replication or the "Sync now" message, as appropriate for the replication type.

Creating and Editing Replication Actions using the CLI

1. The same progress information can be displayed in the CLI, with the state sending shown for a replication in progress:
Creating and Editing Replication Actions using the CLI

2. **Note:** Replication can take a long time to complete, depending on the size of the data being replicated. Use the progress information to determine the update status. For initial replication, it is important to not interrupt it, including, but not limited to, restarting the appliance or canceling the update; otherwise, the entire initial replication must be restarted.

3. **Replication target information can be displayed in the CLI with the state actions:**

```bash
otoro:configuration services replication targets> show
Targets:
   TARGET    LABEL    ACTIONS
   target-000 oakmeal  1
```

```bash
otoro:configuration services replication target-000> select target-000
otoro:configuration services replication target-000> show
Properties:
   address = 10.153.34.167:216
   label = oakmeal
   hostname = oakmeal-7320-167
   asn = 4913649f-7549-6d2a-866b-987d9b4e163
   actions = 1
```

```bash
oakmeal-7320-167:configuration services replication target-000> actions
POOL       PROJECT      SHARE
pool1      project1     (multiple)
```

4. **When using the CLI, it can be helpful to know the ID of the newly created replication action. The ID is used later to select the correct replication action node. To view the ID of the newly created action, use the `last` command to**
navigate to the node with the new replication action. Then use the `get id` command to retrieve the action ID:

```
otoro:> shares
otoro:shares> select p1
otoro:shares p1> replication
otoro:shares p1 replication> create
otoro:shares p1 action (uncommitted)> set target=oakmeal
   target = oakmeal (uncommitted)
otoro:shares p1 action (uncommitted)> set pool=p
   pool = p (uncommitted)
otoro:shares p1 action (uncommitted)> set use_ssl=false
   use_ssl = false (uncommitted)
otoro:shares p1 action (uncommitted)> commit
otoro:shares p1 replication> last get id
   id = fb1bb3fd-3361-42e1-e4a1-b06c426172fb
otoro:shares p1 action-001> done
otoro:shares p1 replication>
```

**Managing Replication Packages**

Packages are containers for replicated projects and shares. Each replication action on a source appliance corresponds to one package on the target appliance as described above. Both the BUI and CLI enable administrators to browse replicated projects, shares, snapshots, and properties much like local projects and shares. However, because replicated shares must exactly match their counterparts on the source appliance, many management operations are not allowed inside replication packages, including creating, renaming, and destroying projects and shares, creating and renaming snapshots, and modifying most properties of projects and shares. Snapshots other than those used as the basis for incremental replication can be destroyed in replication packages. This practice is not recommended but can be used when additional free space is necessary.

In 2009.Q3 and earlier software versions, properties could not be changed on replicated shares. The 2010.Q1 release (with associated deferred upgrades) adds limited support for modifying properties of replicated shares to implement differing policies on the source and target appliances. Such property modifications persist across replication updates. Only the following properties of replicated projects and shares may be modified:

- **Reservation, Compression, Copies, Deduplication, and Caching** - These properties can be changed on the replication target to effect different cost, flexibility, performance, or reliability policies on the target appliance from the source.

- **Mountpoint and Sharing Properties** (e.g., sharens, SMB resource name, etc.) - These properties control how shares are exported to NAS clients and can be changed to effect different security or protection policies on the target appliance from the source.

- **Automatic Snapshot Policies** - Automatic snapshot policies can be changed on the target system but these changes have no effect until the package is severed. Automatic snapshots are not taken or destroyed on replicated projects and shares.
The BUI and CLI don't allow administrators to change immutable properties. For shares, a different icon is used to indicate that the property's inheritance cannot be changed:

**FIGURE 45** Managing Replication Package Properties

Deferred updates provided with the 2010.Q1 release must be applied on replication targets in order to modify properties on such targets. The system will not allow administrators to modify properties inside replication packages on systems which have not applied the 2010.Q1 deferred updates.

The current release does not support configuration of "chained" replication (that is, replicating replicated shares to another appliance).

**Managing Replication Packages using the BUI**

Replication packages are displayed in the BUI as projects under the "Replica" filter:

**FIGURE 46** Replica Filter

Selecting a replication package for editing brings the administrator to the Shares view for the package's project. From here, administrators can manage replicated shares much like local
shares with the exceptions described above. Package properties (including status) can be modified under the Replication tab:

**FIGURE 47** Shares View for the Package's Project

![Shares View for the Package's Project](image)

The status icon on the left changes when replication has failed:

**FIGURE 48** Status Icon Indicates Failure

![Status Icon Indicates Failure](image)

Packages are only displayed in the BUI after the first replication update has begun. They may not appear in the list until some time after the first update has completed.

**Managing Replication Packages using the CLI**

Replication packages are organized in the CLI by source under *shares replication sources*. Administrators first select a source, then a package. Package-level operations can be performed on this node, or the project can be selected to manage project properties and shares just like local projects and shares with the exceptions described above. For example:

```
loader:> shares replication sources
```
Creating and Editing Replication Actions using the CLI

Working with Shares

```plaintext
loader:shares replication sources> show
Sources:

source-000 ayu

<table>
<thead>
<tr>
<th>PROJECT</th>
<th>STATE</th>
<th>LAST UPDATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>package-000 oldproj</td>
<td>idle</td>
<td>unknown</td>
</tr>
<tr>
<td>package-001 aproj1</td>
<td>receiving</td>
<td>Sun Feb 21 2010 22:04:35 GMT+0000 (UTC)</td>
</tr>
</tbody>
</table>

loader:shares replication sources> select source-000
loader:shares replication source-000> select package-001
loader:shares replication source-000 package-001> show
Properties:

enabled = true
state = receiving
state_description = Receiving update
last_sync = Sun Feb 21 2010 22:04:40 GMT+0000 (UTC)
last_try = Sun Feb 21 2010 22:04:40 GMT+0000 (UTC)

Projects:

aproj1

loader:shares replication source-000 package-001> select aproj1
loader:shares replication source-000 package-001 aproj1> get mountpoint
mountpoint = /export
loader:shares replication source-000 package-001 aproj1> get sharenfs
sharenfs = on

You can display replication sources from configuration services replication also. For example:

loader:configuration services replication> show
Properties:

<status> = online

Children:

targets => Configure replication targets
sources => View and manage replication packages
```

Canceling Replication Updates

To cancel in-progress replication updates on the target using the BUI, navigate to the replication package (see above), then click the Replication tab. If an update is in progress, you will see a barber pole progress bar with a cancel button (🚫) next to it as shown here:
Creating and Editing Replication Actions using the CLI

FIGURE 49  Canceling Replication

Click this button to cancel the update.

To cancel in-progress replication updates on the target using the CLI, navigate to the replication package (see above) and use the cancelupdate command.

It is not possible to initiate updates from the target. Administrators must login to the source system to initiate a manual update.

Disabling a Replication Package

Replication updates for a package can be disabled entirely, cancelling any ongoing update and causing new updates from the source appliance to fail.

To toggle whether a package is disabled from the BUI, navigate to the package (see above), then click the Replication tab, and then click the icon. The status icon on the left should change to indicate the package's status (enabled, disabled, or failed). The package remains disabled until explicitly enabled by an administrator using the same button or the CLI.

To toggle whether a package is disabled from the CLI, navigate to the package (see above), modify the enabled property, and commit your changes.

Cloning a Replication Package or Share

A clone of a replicated package is a local, mutable project that can be managed like any other project on the system. The clone's shares are clones of the replicated shares at the most recently received snapshot. These clones share storage with their origin snapshots in the same way as clones of share snapshots do. This mechanism can be used to failover in the case of a catastrophic problem at the replication source, or simply to provide a local version of the data that can be modified.
Use the button in the BUI or the \texttt{clone} CLI command (in the package's context) to create a package clone based on the most recently received replication snapshot. Both the CLI and BUI interface require the administrator to specify a name for the new clone project and allow the administrator to override the mountpoint of the project or its shares to ensure that they don't conflict with those of other shares on the system.

In 2009.Q3 and earlier, cloning a replicated project was the only way to access its data and thus the only way to implement disaster-recovery failover. In 2010.Q1 and later, individual filesystems can be exported read-only without creating a clone. Additionally, replication packages can be directly converted into writable local projects as part of a failover operation. As a result, cloning a package is no longer necessary or recommended, as these alternatives provide similar functionality with simpler operations and without having to manage clones and their dependencies.

In particular, while a clone exists, its origin snapshot cannot be destroyed. When destroying the snapshot (possibly as a result of destroying the share, project, or replication package of which the snapshot is a member), the system warns administrators of any dependent clones which will be destroyed by the operation. Note that snapshots can also be destroyed on the source at any time and such snapshots are destroyed on the target as part of the subsequent replication update. If such a snapshot has clones, the snapshot will instead be renamed with a unique name (typically \texttt{recv-XXX}).

Administrators can also clone individual replicated share snapshots using the normal BUI and CLI interfaces.

**Exporting Replicated Filesystems**

Replicated filesystems can be exported read-only to NAS clients. This can be used to verify the replicated data or to perform backups or other intensive operations on the replicated data (offloading such work from the source appliance).

The filesystem's contents always match the most recently received replication snapshot for that filesystem. This may be newer than the most recently received snapshot for the entire package, and it may not match the most recent snapshot for other shares in the same package. See “Understanding Replication Snapshots and Data Consistency” on page 346 for details.

Replication updates are applied atomically at the filesystem level. Clients looking at replicated files will see replication updates as an instantaneous change in the underlying filesystem. Clients working with files deleted in the most recent update will see errors. Clients working with files changed in the most recent update will immediately see the updated contents.

Replicated filesystems are not exported by default. They are exported by modifying the "exported" property of the project or share using the BUI or CLI:
This property is inherited like other share properties. This property is not shown for local projects and shares because they are always exported. Additionally, severing replication (which converts the package into a local project) causes the package's shares to become exported.

Replicated LUNs currently cannot be exported. They must be first cloned or the replication package severed in order to export their contents.

**Severing Replication**

A replication package can be converted into a local, writable project that behaves just like other local projects (i.e. without the management restrictions applied to replication packages) by severing the replication connection. After this operation, replication updates can no longer be received into this package, so subsequent replication updates of the same project from the source will need to send a full update with a new action (into a new package). Subsequent replication updates using the same action will fail because the corresponding package no longer exists on the target.

This option is primarily useful when using replication to migrate data between appliances or in other scenarios that don't involve replicating the received data back to the source as part of a typical two-system disaster recovery plan.

Replication can be severed from the BUI by navigating to the replication package (see above), clicking the Replication tab, and clicking the button. The resulting dialog allows the administrator to specify the name of the new local project.

Replication can be severed from the CLI by navigating to the replication package (see above), and using the `sever` command. This command takes an optional argument specifying the name of the new local project. If no argument is specified, the original name is used.

Because all local shares are exported, all shares in a package are exported when the package is severed, whether or not they were previously exported (see above). If there are mountpoint conflicts between replicated filesystems and other filesystems on the system, the sever operation will fail. These conflicts must be resolved before severing by reconfiguring the mountpoints of the relevant shares.
Reversing the Direction of Replication

The direction of the replication can be reversed to support typical two-system disaster recovery plans. This operation is similar to the sever operation described above, but additionally configures a replication action on the new local project for incremental replication back to the source system. No changes are made on the source system when this operation is completed, but the first update attempt using this action will convert the original project on the source system into a replication package and rollback any changes made since the last successful replication update from that system.

This feature does not automatically redirect production workloads, failover IP addresses, or perform other activities related to the disaster-recovery failover besides modifying the read-write status of the primary and secondary data copies.

As part of the conversion of the original source project into a replication package on the original source system (now acting as the target), the shares that were replicated as part of the action/package currently being reversed are moved into a new replication package and unexported. The original project remains in the local collection but may end up empty if the action/package included all of its shares. When share-level replication is reversed, any other shares in the original project remain unchanged.

After establishing share-level replication from one appliance to another, reversing that replication on the target appliance destroys the replication schedule. A replication action is then created at the project level which contains the correct target appliance without a schedule.

As mentioned above, this feature is typically used to implement a two-system disaster recovery configuration in which a primary system serves production data and replicates it to a secondary or DR system (often in another data center) standing by to take over the production traffic in the event of a disaster at the primary site. In the event of a disaster at the primary site, the secondary site's copy must be made "primary" by making it writable and redirecting production traffic to the secondary site. When the primary site is repaired, the changes accumulated at the secondary site can be replicated back to the primary site and that site can resume servicing the production workload.

A typical sequence of events under such a plan is as follows:

- The primary system is serving the production workload and replicating to the secondary system.
- A disaster occurs, possibly representing a total system failure at the primary site. Administrators reverse the direction of replication on the secondary site, exporting the replicated shares under a new project configured for replication back to the primary site for when primary service is restored. In the meantime, the production workload is redirected to the secondary site.
- When the primary site is brought back online, an administrator initiates a replication update from the secondary site to the primary site. This converts the primary's copy into a replication package, rolling back any changes made since the last successful update to the target (before the failure). When the primary site's copy is up-to-date, the administrator
Reversing Replication using the BUI

reverses the direction of replication again, making the copy at the primary site writable. Production traffic is redirected back to the primary site. Replication is resumed from the primary to the secondary, restoring the initial relationship between the primary and secondary copies.

When reversing the direction of replication for a package, it is strongly recommended that administrators first stop replication of that project from the source. If a replication update is in progress when an administrator reverses the direction of replication for a project, administrators cannot know which consistent replication snapshot was used to create the resulting project on the former target appliance (now source appliance).

Replication can be reversed from the BUI by navigating to the replication package (see above), clicking the Replication tab, and clicking the button. The resulting dialog allows the administrator to specify the name of the new local project.

Replication can be reversed from the CLI by navigating to the replication package (see above), and using the reverse command. This command takes an optional argument specifying the name of the new local project. If no argument is specified, the original name is used.

Because all local shares are exported, all shares in a package are exported when the package is reversed, whether or not they were previously exported (see above). If there are mount point conflicts between replicated filesystems and other filesystems on the system, the reverse operation will fail. These conflicts must be resolved before severing by reconfiguring the mount points of the relevant shares. Because this operation is typically part of the critical path of restoring production service, it is strongly recommended to resolve these mount point conflicts when the systems are first set up rather than at the time of DR failover.

Destroying a Replication Package

The project and shares within a package cannot be destroyed without destroying the entire package. The entire package can be destroyed from the BUI by destroying the corresponding project. A package can be destroyed from the CLI using the destroy command at the node.

When a package is destroyed, subsequent replication updates from the corresponding action will fail. To resume replication, the action will need to be recreated on the source to create a new package on the target into which to receive a new copy of the data.

Reversing Replication using the BUI

The following is an example of reversing replication to support typical two-system disaster recovery. In this example M11 is the production system and M5 is the recovery system.
1. On Production System M11, navigate to Configuration > SERVICES.

2. On the SMB line under Data Services, if the status is Disabled, click Enable service.


4. Click Targets and configure name, hostname and password settings. Name=M5, Host name=192.168.1.17, Root password=pppp$1234

5. Select Pool=Pool1.

6. Navigate to Shares > PROJECTS.

7. Click Projects. Name=P1

8. Navigate to Shares > PROJECTS > P1 > Protocols.

9. In the SMB section, set Resource Name=on.

10. Navigate to Shares > PROJECTS > P1 > Shares.

11. Click FileSystems. Name=S1, User=root, Group=other, Permissions=RWX RWX RWX

12. Navigate to Shares > PROJECTS > P1 > Shares > S1 > Protocols. The SMB section shows that S1 can be reached using SMB at \192.168.1.7\S1.

13. Navigate to Shares > PROJECTS > P1 > Replication.

14. Click Actions and set the target and pool. Target=M5, Pool=Pool1

15. Click Schedule and set the frequency. Frequency=Half-Hour at 00 minutes past the hour.

16. On SMB Client System, Map network drive \192.168.1.7\S1 (user=root, password=pppp$1234).

17. Create file F1.txt.

18. On Production System M11, navigate to Shares > PROJECTS > P1 > Replication.

19. On the Action line TARGET=M5, click Update now.
20. After replication is finished, click Disable, STATUS changes to disabled.

Reversing Replication for Disaster Recovery using the BUI

To simulate recovery from a disaster that prevents access to the production system, use the recovery system to reverse replication. When replication is reversed, the replication package present on the target is converted to a local project and additionally a replication action is configured for this local project for incremental replication back to the original source system. This replication action is not enabled by default. The administrator should send the update manually.

1. To simulate loss of contact with Primary System M11 on the SMB Client System, select Disconnect network drive.

2. On Disaster Recovery System M5 select Pool=Pool1.

3. Navigate to Shares > PROJECTS > REPLICA. The Project M11:P1 is listed

4. Navigate to Shares > PROJECTS > REPLICA> M11:P1 > Replication. The package has the Status=Idle

5. Click Reverse the direction of replication and set the new project name. New Project Name=P1

6. Navigate to Shares > PROJECTS > REPLICA. Project M11:P1 is not listed anymore because the replication package present on the target is converted to a local project.

7. Navigate to Configuration > SERVICES.

8. On the SMB line under Data Services, if the status is Disabled, click Enable service.

9. Navigate to Shares > PROJECTS > LOCAL. Project P1 is listed.


11. In the SMB section, set Resource Name=on.

12. Navigate to Shares > PROJECTS > P1 > Shares > S1 > Protocols. The SMB section shows that S1 can be reached using SMB at \192.168.1.17\S1.
13. On SMB Client System, Map network drive \192.168.1.17\S1 (user=root, password=pppp$1234).

14. Edit file F1.txt and save it as F2.txt. NOTE: During a real disaster recovery sequence, once communications with Production System M11 were restored, you would have the opportunity to trigger manual, scheduled, or continuous replication updates while applications continue to access data on Disaster Recovery System M5.

15. To prepare for the transition back to the production system, select Disconnect network drive.

16. On Disaster Recovery System M5, navigate to Shares > PROJECTS > P1 > Replication.

17. On Action line TARGET=M11 click Update now.

18. After replication is finished, click Disable.

Reversing Replication to Resume Replication from Production System using the BUI

Each time replication is reversed, you provide a new project name that is used when the replication package is converted into a new local project. If you want to use the same name and a prior replication reversal has left behind an empty local project with that name, you must delete the existing empty project so that the next reversal can create a project with the same name.

1. On Production System M11, navigate to Shares > PROJECTS > LOCAL > P1. P1 is empty because the first update from the new source system (the original target) converts the original project on the original source system into a replication package. This example employs a project-level replication action that replicates all of the shares in the project. Therefore, all of the shares that were present under this local project are now present under the replication package, leaving the local project empty.

2. Navigate to Shares > PROJECTS > LOCAL.

3. Delete P1. It is safe to delete this empty project because its contents have been moved to the replication package as a result of the replication reversal from the original target.

4. Navigate to Shares > PROJECTS > REPLICA > M5:P1 > Replication.
5. Click Reverse the direction of replication and set the project name. New Project Name=P1

6. On SMB Client System, Map network drive \192.168.1.7\S1 (user=root, password=pppp$1234). Files F1.txt and F2.txt are listed.

7. On Production System M11, navigate to Shares > PROJECTS > P1 > Replication.

8. On Action line TARGET=M5, click Edit entry.

9. Click Schedule and set the frequency. Frequency=Half-Hour at 00 minutes past the hour.

10. On Action line TARGET=M5, click Update now. This directs Disaster Recovery System M5 to convert the its local project P1 back into replication package M11:P1.

11. Monitor the UPDATES column on action line TARGET=M5 and wait for the replication update to complete.

12. On Disaster Recovery System M5, navigate to Shares > PROJECTS > LOCAL > P1. P1 is empty because the project was converted back into a replication package.

13. Navigate to Shares > PROJECTS > LOCAL.

14. To enable the next reversal to convert the replication package to a project named P1, delete P1.

Forcing Replication to use a Static Route using the BUI

To consolidate replication traffic onto a specific network interface, you must connect the source and target ZFS appliances using static routes. To set up the static routes, use the following steps:

1. To set up a static route, on the Configuration > Network > Routing page, click the add icon.

2. In the Insert Static Route box, select the Family and Kind and then enter the Destination IP, Gateway, and Interface.
3. Click Add.

4. To ensure traffic is routing on the source and target, in the CLI use `traceroute`. For information about using `traceroute`, see “Configuring Network Routing” on page 63. In the example, `10.80.219.124 @ igb0` identifies `igb0` as the interface. This is a quick way to verify the right interface is being used.

```
brmv01sn02:> traceroute poc7330-050
traceroute: Warning: Multiple interfaces found; using 10.80.219.124 @ igb0 traceroute to poc7330-050 (10.80.219.117), 30 hops max, 40 byte packets
  1  poc7330-050.us.oracle.com (10.80.219.117)  0.446 ms  0.115 ms  0.104 ms
```

5. To add a new replication target, on the Configuration > Services > Replication page, click the `add` icon.

6. In the Add Replication Target box, type a name for the target, the Hostname IP address for the network interface, and the password.
Forcing Replication to use a Static Route using the BUI

7. Click Add.

8. To ensure traffic passes across the defined static route, after replication starts use Network Interface Bytes.

For more information, see “Network: Interface Bytes” in “Oracle ZFS Storage Appliance Analytics Guide”.

Note - To use Advanced Analytics, on the Configuration > Preferences page ensure "Make available advanced analytics statistics" is toggled on.
9. After you verify that source to target replication uses the correct interface, reverse the replication.

For information about reversing replication, see “Reversing the Direction of Replication” on page 365.

Cloning a Received Replication Project using the CLI

This is a CLI example of cloning a received replication project, overriding both the project's and one share's mountpoint:

- To clone a received replication project, use the following CLI commands.

```
perch:> shares
perch:shares> replication
perch:shares replication> sources
perch:shares replication sources> select source-000
perch:shares replication source-000> select package-000
perch:shares replication source-000 package-000> clone
perch:shares replication source-000 package-000 clone> set target_project=my_clone
   target_project = my_clone
perch:shares replication source-000 package-000 clone> list
CLONE PARAMETERS
   target_project = my_clone
   original_mountpoint = /export
   override_mountpoint = false
   mountpoint =
SHARE                           MOUNTPOINT
   bob                             (inherited)
   myfs1                           (inherited)
perch:shares replication source-000 package-000 clone> set override_mountpoint=true
   override_mountpoint = true
perch:shares replication source-000 package-000 clone> set mountpoint=/export/my_clone
   mountpoint = /export/my_clone
perch:shares replication source-000 package-000 clone> select bob
perch:shares replication source-000 package-000 clone bob> set override_mountpoint=true
   override_mountpoint = true
perch:shares replication source-000 package-000 clone bob> set mountpoint=/export/bob
   mountpoint = /export/bob
perch:shares replication source-000 package-000 clone bob done
perch:shares replication source-000 package-000 clone> commit
CLONE PARAMETERS
   target_project = my_clone
   original_mountpoint = /export
   override_mountpoint = true
   mountpoint = /export/my_clone
```
**Working with Shadow Migration**

A common task for administrators is to move data from one location to another. In the most abstract sense, this problem encompasses a large number of use cases, from replicating data between servers to keeping user data on laptops in sync with servers. There are many external tools available to do this, but the appliance has two integrated solutions for migrating data that addresses the most common use cases. The first, replication, is intended for replicating data between one or more appliances, and is covered separately; see “Working with Replication” on page 337. The second, shadow migration, is described here.

Shadow migration is a process for migrating data from external NAS sources with the intent of replacing or decommissioning the original once the migration is complete. This is most often used when introducing a new appliance into an existing environment in order to take over file sharing duties of another server, but a number of other novel uses are possible, outlined below.

To use shadow migration, see the following sections:

- “Understanding Shadow Migration” on page 374
- “Creating a Shadow Filesystem” on page 377
- “Managing Background Migration” on page 377
- “Handling Migration Errors” on page 378
- “Monitoring Migration Progress” on page 378
- “Canceling Migration” on page 379
- “Snapshotting Shadow File Systems” on page 379
- “Backing Up Shadow File Systems” on page 379
- “Replicating Shadow File Systems” on page 380
- “Migrating Local File Systems” on page 380
- “Using Shadow Migration Analytics” on page 380
- “Testing Potential Shadow Migration using the CLI” on page 381
- “Migrating Data from an Active NFS Server using the CLI” on page 381

**Understanding Shadow Migration**

Shadow migration uses interposition, but it is integrated into the appliance and does not require a separate physical machine. When shares are created, they can optionally "shadow" an existing directory, either locally or over NFS. In this scenario, downtime is scheduled once, where the
source appliance X is placed into read-only mode, a share is created with the shadow property set, and clients are updated to point to the new share on the Oracle ZFS Storage Appliance. Clients can then access the appliance in read-write mode.

**FIGURE 51**  Shadow Migration

Once the shadow property is set, data is transparently migrated in the background from the source appliance locally. If a request comes from a client for a file that has not yet been migrated, the appliance will automatically migrate this file to the local server before responding to the request. This may incur some initial latency for some client requests, but once a file has been migrated, all accesses are local to the appliance and have native performance. It is often the case that the current working set for a filesystem is much smaller than the total size, so once this working set has been migrated, regardless of the total native size on the source, there will be no perceived impact on performance.

The downside to shadow migration is that it requires a commitment before the data has finished migrating, though this is the case with any interposition method. During the migration, portions of the data exist in two locations, which means that backups are more complicated,
and snapshots may be incomplete and/or exist only on one host. It is therefore extremely important that any migration between two hosts first be tested thoroughly to make sure that identity management and access controls are setup correctly. This need not test the entire data migration, but it should be verified that files or directories that are not world readable are migrated correctly, ACLs (if any) are preserved, and identities are properly represented on the new system.

Shadow migration is implemented using on-disk data within the filesystem, so there is no external database and no data stored locally outside the storage pool. If a pool is failed over in a cluster, or both system disks fail and a new head node is required, all data necessary to continue shadow migration without interruption will be kept with the storage pool.

The following lists the restrictions on the shadow source:

- In order to properly migrate data, the source filesystem or directory *must be read-only*. Changes made to files source may or may not be propagated based on timing, and changes to the directory structure can result in unrecoverable errors on the appliance.
- Shadow migration supports migration only from NFS sources. NFSv4 shares will yield the best results. NFSv2 and NFSv3 migration are possible, but ACLs will be lost in the process and files that are too large for NFSv2 cannot be migrated using that protocol. Migration from SMB sources is not supported.
- Shadow migration of LUNs is not supported.

During migration, if the client accesses a file or directory that has not yet been migrated, there is an observable effect on behavior. The following lists the shadow file system semantics:

- For directories, client requests are blocked until the entire directory is migrated. For files, only the portion of the file being requested is migrated, and multiple clients can migrate different portions of a file at the same time.
- Files and directories can be arbitrarily renamed, removed, or overwritten on the shadow filesystem without any effect on the migration process.
- For files that are hard links, the hard link count may not match the source until the migration is complete.
- The majority of file attributes are migrated when the directory is created, but the on-disk size (st_nblocks in the UNIX stat structure) is not available until a read or write operation is done on the file. The logical size will be correct, but a du(1) or other command will report a zero size until the file contents are actually migrated.
- If the appliance is rebooted, the migration will pick up where it left off originally. While it will not have to re-migrate data, it may have to traverse some already-migrated portions of the local filesystem, so there may be some impact to the total migration time due to the interruption.
- Data migration makes use of private extended attributes on files. These are generally not observable except on the root directory of the filesystem or through snapshots. Adding, modifying, or removing any extended attribute that begins with SUNWshadow will have undefined effects on the migration process and will result in incomplete or corrupt state. In addition, filesystem-wide state is stored in the .SUNWshadow directory at the root of the filesystem. Any modification to this content will have a similar effect.
Once a filesystem has completed migration, an alert will be posted, and the shadow attribute will be removed, along with any applicable metadata. After this point, the filesystem will be indistinguishable from a normal filesystem.

Data can be migrated across multiple filesystems into a single filesystem, through the use of NFSv4 automatic client mounts (sometimes called "mirror mounts") or nested local mounts.

Use the following rules to migrate identity information for files, including ACLs:

- The migration source and target appliance must have the same name service configuration.
- The migration source and target appliance must have the same NFSv4 mapid domain.
- The migration source must support NFSv4. Use of NFSv3 is possible, but some loss of information will result. Basic identity information (owner and group) and POSIX permissions will be preserved, but any ACLs will be lost.
- The migration source must be exported with root permissions to the appliance.

If you see files or directories owned by "nobody", it likely means that the appliance does not have name services setup correctly, or that the NFSv4 mapid domain is different. If you get 'permission denied' errors while traversing filesystems that the client should otherwise have access to, the most likely problem is failure to export the migration source with root permissions.

Creating a Shadow Filesystem

The shadow migration source can only be set when a filesystem is created. In the BUI, this is available in the filesystem creation dialog. In the CLI, it is available as the shadow property. The property takes one of the following forms:

- **Local** - file:///<path>
- **NFS** - nfs://<host>/<path>

The BUI also allows the alternate form <host>:/<path> for NFS mounts, which matches the syntax used in UNIX systems. The BUI also sets the protocol portion of the setting (file:// or nfs://) via the use of a pull down menu. When creating a filesystem, the server will verify that the path exists and can be mounted.

Managing Background Migration

When a share is created, it will automatically begin migrating in the background, in addition to servicing inline requests. This migration is controlled by the shadow migration service. There is a single global tunable, which is the number of threads dedicated to this task. Increasing the number of threads will result in greater parallelism at the expense of additional resources.

The shadow migration service can be disabled, but this should only be used for testing purposes, or when the active of shadow migration is overwhelming the system to the point
where it needs to be temporarily stopped. When the shadow migration service is disabled, synchronous requests are still migrated as needed, but no background migration occurs. With the service disabled, no shadow migration will ever complete, even if all the contents of the filesystem are read manually. It is highly recommended to always leave the service enabled.

**Handling Migration Errors**

Because shadow migration requires committing new writes to the server prior to migration being complete, it is very important to test migration and monitor for any errors. Errors encountered during background migration are kept and displayed in the BUI as part of shadow migration status. Errors encountered during other synchronous migration are not tracked, but will be accounted for once the background process accesses the affected file. For each file, the remote filename as well as the specific error are kept. Clicking on the information icon next to the error count will bring up this detailed list. The error list is not updated as errors are fixed, but simply cleared by virtue of the migration completing successfully.

Shadow migration will not complete until all files are migrated successfully. If there are errors, the background migration will continually retry the migration until it succeeds. This allows the administrator to fix any errors (such as permission problems), let the migration complete, and be assured of success. If the migration cannot complete due to persistent errors, the migration can be canceled, leaving the local filesystem with whatever data was able to be migrated. This should only be used as a last resort; once migration has been canceled, it cannot be resumed.

**Monitoring Migration Progress**

To monitor shadow migration progress, the appliance provides such statistics as:

- Size of data transferred so far
- Estimate of remaining size to be migrated
- Migration time so far
- Migration errors

At the beginning of migration, the appliance obtains the source filesystem statistics and calculates its size. It uses these values to provide a reasonably accurate visual representation of migration progress and an estimation of the remaining data to be migrated. Of note, the remaining bytes is an estimate based on the assumption that an entire filesystem is being migrated. If only part of the source file system is migrated, the remaining bytes estimate is inaccurate. If the source filesystem has nested filesystems, the total filesystem size is recalculated when the nested mount point is discovered during migration, and the remaining bytes are re-estimated based on the newly calculated total. Estimation of remaining bytes may be inaccurate if the source filesystem uses compression. These values are available in the BUI and CLI through both the standard filesystem properties as well as properties of the shadow migration node (or UI panel).
Canceling Migration

Migration can be canceled, but this should only be done in extreme circumstances when the source is no longer available. Once migration has been canceled, it cannot be resumed. The primary purpose is to allow migration to complete when there are uncorrectable errors on the source. If the complete filesystem has finished migration except for a few files or directories, and there is no way to correct these errors (i.e. the source is permanently broken), then canceling the migration will allow the local filesystem to resume status as a ‘normal’ filesystem.

To cancel migration in the BUI, click the close icon next to the progress bar in the left column of the share in question. In the CLI, migrate to the shadow node beneath the filesystem and run the cancel command.

Snapshotting Shadow File Systems

Shadow filesystems can be snapshotted; however, the state of what is included in the snapshot is arbitrary. Files that have not yet been migrated will not be present, and implementation details (such as SUNWshadow extended attributes) may be visible in the snapshot. This snapshot can be used to restore individual files that have been migrated or modified since the original migration began. Because of this, it is recommended that any snapshots be kept on the source until the migration is completed, so that unmigrated files can still be retrieved from the source if necessary. Depending on the retention policy, it may be necessary to extend retention on the source in order to meet service requirements. While snapshots can be taken, these snapshots cannot be rolled back to, nor can they be the source of a clone.

Backling Up Shadow File Systems

Filesystems that are actively migrating shadow data can be backed using NDMP as with any other filesystem. The shadow setting is preserved with the backup stream, but will be restored only if a complete restore of the filesystem is done and the share doesn't already exist. Restoring individual files from such a backup stream or restoring into existing filesystems may result in inconsistent state or data corruption. During the full filesystem restore, the filesystem will be in an inconsistent state (beyond the normal inconsistency of a partial restore) and shadow migration will not be active. Only when the restore is completed is the shadow setting restored. If the shadow source is no longer present or has moved, the administrator can observe any errors and correct them as necessary.
Replicating Shadow File Systems

Filesystems that are actively migrating shadow data can be replicated using the normal mechanism, but only the migrated data is sent in the data stream. As such, the remote side contains only partial data that may represent an inconsistent state. The shadow setting is sent along with the replication stream, so when the remote target is failed over, it will keep the same shadow setting. As with restoring an NDMP backup stream, this setting may be incorrect in the context of the remote target. After failing over the target, the administrator can observe any errors and correct the shadow setting as necessary for the new environment.

Migrating Local File Systems

In addition to its primary purpose of migrating data from remote sources, the same mechanism can also be used to migrate data from local filesystem to another on the appliance. This can be used to change settings that otherwise cannot be modified, such as creating a compressed version of a filesystem or changing the recordsizer for a filesystem after the fact. In this model, the old share (or subdirectory within a share) is made read-only or moved aside, and a new share is created with the shadow property set using the file protocol. Clients access this new share, and data is written using the settings of the new share.

Using Shadow Migration Analytics

In addition to standard monitoring on a per-share basis, it is also possible to monitor shadow migration system-wide through Analytics. The shadow migration analytics are available under the "Data Movement" category. There are three basic statistics available:

- **Shadow Migration Requests** - This statistic tracks requests for files or directories that are not cached and known to be local to the filesystem. It does account for both migrated and unmigrated files and directories, and it can be used to track the latency incurred as part of shadow migration, as well as track the progress of background migration. It can be broken down by file, share, project, or latency. It currently encompasses both synchronous and asynchronous (background) migration, so it is not possible to view only latency visible to clients.

- **Shadow Migration Bytes** - This statistic tracks bytes transferred as part of migrating file or directory contents. This does not apply to metadata (extended attributes, ACLs, etc.). It gives a rough approximation of the data transferred, but source datasets with a large amount of metadata will show a disproportionally small bandwidth. The complete bandwidth can be observed by looking at network analytics. This statistic can be broken down by local filename, share, or project.

- **Shadow Migration Operations** - This statistic tracks operations that require going to the source filesystem. This can be used to track the latency of requests from the shadow migration source. It can be broken down by file, share, project, or latency.
Testing Potential Shadow Migration using the CLI

Before attempting a complete migration, it is important to test the migration to make sure that the appliance has appropriate permissions and security attributes are translated correctly. Once you are confident that the basic setup is functional, the shares can be setup for the final migration.

Note - As part of capacity planning, remember to take into account default/user group quotas because the quotas could be exceeded if the source is larger than the destination. Also, shadow migration will fail if the target runs out of disk space.

1. Configure the source so that the appliance has root access to the share. This typically involves adding an NFS host-based exception or setting the anonymous user mapping (the latter having more significant security implications).

2. Create a share on the local filesystem with the shadow attribute set to 'nfs://<host>/<snapshotpath>' in the CLI or just '<host>/<snapshotpath>' in the BUI (with the protocol selected as 'NFS'). The snapshot should be read-only copy of the source. If no snapshots are available, a read-write source can be used, but may result in undefined errors.

3. Validate that file contents and identity mapping are correctly preserved by traversing the file structure.

4. If the data source is read-only (as with a snapshot), let the migration complete and verify that there were no errors in the transfer.

Migrating Data from an Active NFS Server using the CLI

1. Schedule downtime during which clients can be quiesced and reconfigured to point to a new server.

2. Configure the source so that the appliance has root access to the share. This typically involves adding an NFS host-based exception or setting the anonymous user mapping (the latter having more significant security implications).

3. Configure the source to be read-only. This step is technically optional, but it is much easier to guarantee compliance if it is impossible for misconfigured clients to write to the source while migration is in progress.
4. Create a share on the local filesystem with the shadow attribute set to "nfs://<host>/<path>" in the CLI or just '<host>/<path>' in the BUI (with the protocol selected as 'NFS').

5. Reconfigure clients to point at the local share on the appliance.

   At this point, shadow migration should be running in the background, and client requests should be serviced as necessary. You can observe the progress as described above. Multiple shares can be created during a single scheduled downtime through scripting the CLI.

Working with Projects

Shares, filesystems and LUNs can be grouped into projects. A project defines a common administrative control point for managing shares. Shares within a project can share common settings, and quotas can be enforced at the project level in addition to the share level. Projects can also be used solely for grouping logically related shares together, so their common attributes (such as accumulated space) can be accessed from a single point.

By default, the appliance creates a single default project when a storage pool is first configured. It is possible to create all shares within this default project, although for reasonably sized environments creating additional projects is strongly recommended, if only for organizational purposes.

To work with projects, see the following sections:

- "Managing Projects using the BUI" on page 382
- "Project Usage Statistics" on page 383
- "Creating Projects using the BUI" on page 384
- "Navigating Projects using the CLI" on page 384
- "Managing Projects using the CLI" on page 385
- "Selecting a Cluster Pool using the CLI" on page 386
- "Project CLI Properties" on page 387
- "Shares Project General BUI Page" on page 388
- "Project Access" on page 389
- "Project Snapshots" on page 390

Managing Projects using the BUI

The Projects UI is accessed from "Shares -> Projects". This presents a list of all projects on the system, although projects can be selected by using the project panel or by clicking the project name while editing a share within a project.
After navigating to the project view, you will be presented with a list of projects on the system. Alternatively, you can navigate to the shares screen and open the project panel for a shortcut to projects. The panel does not scale well to large numbers of projects, and is not a replacement for the complete project list. The following fields are displayed for each project:

**TABLE 140  Project Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the share. The share name is an editable text field. Clicking on the name will allow you to enter a new name for the project. Hitting return or moving focus from the name will commit the change. You will be asked to confirm the action, as renaming shares requires disconnecting active clients.</td>
</tr>
<tr>
<td>Size</td>
<td>The total size of all shares within the project and unused reservation.</td>
</tr>
</tbody>
</table>

The following tools are available for each project:

**TABLE 141  Project Icons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="edit_icon" alt="Edit" /></td>
<td>Edit an individual project (also accessible by double-clicking the row).</td>
</tr>
<tr>
<td><img src="delete_icon" alt="Delete" /></td>
<td>Destroy the project. You will be prompted to confirm this action, as it will destroy all data in the share and cannot be undone.</td>
</tr>
</tbody>
</table>

To edit a project, click on the pencil icon or double-click the row in the project list, or click on the name in the project panel. This will select the project, and give several different tabs to choose from for editing properties of the project.

The name of the project is presented in the upper left corner to the right of the project panel. The name of the project can also be changed by clicking on the project name and entering new text into the input. You will be asked to confirm this action, as it will require disconnecting active clients of the project.

The left side of the shares view also shows static properties when editing a particular project. These properties are read only, and cannot be modified. For information about Compression ratio, see “Working with the Shares > Shares BUI Page” on page 295.

**Project Usage Statistics**

On the left side of the view (beneath the project panel when expanded) is a table explaining the current space usage statistics. If any properties are zero, then they are excluded from the table.
The majority of these properties are identical between projects and shares, though there are some statistics that only have meaning for projects.

- **Available space** - See “Working with the Shares > Shares BUI Page” on page 295.
- **Referenced data** - Sum of all referenced data for all shares within the project, in addition to a small amount of project overhead. See “Working with the Shares > Shares BUI Page” on page 295 for more information on how referenced data is calculated for shares.
- **Snapshot data** - Sum of all snapshot data for all shares, and any project snapshot overhead. See “Working with the Shares > Shares BUI Page” on page 295 for more information on how snapshot data is calculated for shares.
- **Unused Reservation** - Unused reservation for the project. This only includes data not currently used for the project level reservation. It does not include unused reservations of any shares contained in the project.
- **Unused Reservation of shares** - Sum of unused reservation of all shares. See “Working with the Shares > Shares BUI Page” on page 295 for more information on how unused reservation is calculated for shares.
- **Total space** - The sum of referenced data, snapshot data, unused reservation, and unused reservation of shares.

### Creating Projects using the BUI

1. To create a project, view the list of projects and click the **+** button.

2. Alternatively, clicking the "Add..." button in the project panel will present the same dialog. Enter the project name and click apply to create the project.

### Navigating Projects using the CLI

The projects CLI is under shares.

- **To select a project, use the `select` command.**

```
clownfish:> shares
clownfish:shares> select default
clownfish:shares default> get
   aclinherit = restricted
   aclmode = discard
   atime = true
   checksum = fletcher4
   compression = off
```
Managing Projects using the CLI

A project is created using the `project` command. The properties can be modified as needed before committing the changes:

1. **To create a project use the `project` command.**

```bash
clownfish:shares> project home
clownfish:shares home (uncommitted)> get
  mountpoint = /export (default)
  quota = 0 (default)
  reservation = 0 (default)
  sharesmb = off (default)
  sharenfs = on (default)
  sharedav = off (default)
  sharedftp = off (default)
  default_group = other (default)
  default_permissions = 700 (default)
```
Selecting a Cluster Pool using the CLI

In an active/active cluster configuration, one node can be in control of both pools while failed over. In this case, the CLI context will show the current pool in parenthesis.

1. To change pools use the `set` command from the top level shares context.

   ```
   clownfish:shares (pool-0)> set pool=pool-1
   clownfish:shares (pool-1)>
   ```

2. Once the pool context has been select, projects and shares are managed within that pool using the standard CLI interfaces.

   ```
   clownfish:shares (pool-1)> commit
   ```

2. To destroy a project use the `destroy` command.

   ```
   clownfish:shares> destroy home
   This will destroy all data in "home"! Are you sure? (Y/N)
   ```

3. To rename a project use the `rename` command.

   ```
   clownfish:shares> rename default home
   ```
## Project CLI Properties

The following properties are available in the CLI, with their equivalent in the BUI. Properties can be set using the standard CLI commands get and set. In addition, properties can be inherited from the parent project by using the unset command.

<table>
<thead>
<tr>
<th>CLI Name</th>
<th>Type</th>
<th>BUI Name</th>
<th>BUI Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>aclinhibit</td>
<td>inherited</td>
<td>Project Access</td>
<td>Access</td>
</tr>
<tr>
<td>aclmode</td>
<td>inherited</td>
<td>Project Access</td>
<td>Access</td>
</tr>
<tr>
<td>atime</td>
<td>inherited</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>checksum</td>
<td>inherited</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>compression</td>
<td>inherited</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>compressratio</td>
<td>read-only</td>
<td>Projects</td>
<td>Static</td>
</tr>
<tr>
<td>copies</td>
<td>inherited</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>creation</td>
<td>read-only</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>dedup</td>
<td>inherited</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>default_group</td>
<td>creation default</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>default_permissions</td>
<td>creation default</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>default_sparse</td>
<td>creation default</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>default_user</td>
<td>creation default</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>default_volblocksize</td>
<td>creation default</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>default_volsize</td>
<td>creation default</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>defaultgroupquota</td>
<td>space management</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>defaultuserquota</td>
<td>space management</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>mountpoint</td>
<td>inherited</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>mbmand</td>
<td>inherited</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>quota</td>
<td>space management</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>readonly</td>
<td>inherited</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>recordsize</td>
<td>inherited</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>reservation</td>
<td>space management</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>secondary cache</td>
<td>inherited</td>
<td>Project - General</td>
<td>General</td>
</tr>
<tr>
<td>sharedav</td>
<td>inherited</td>
<td>Project Protocols</td>
<td>Protocols</td>
</tr>
<tr>
<td>shareftp</td>
<td>inherited</td>
<td>Project Protocols</td>
<td>Protocols</td>
</tr>
<tr>
<td>sharensfs</td>
<td>inherited</td>
<td>Project Protocols</td>
<td>Protocols</td>
</tr>
<tr>
<td>sharesmb</td>
<td>inherited</td>
<td>Project Protocols</td>
<td>Protocols</td>
</tr>
<tr>
<td>snapdir</td>
<td>inherited</td>
<td>Project Snapshots</td>
<td>Snapshots</td>
</tr>
<tr>
<td>snaplabel</td>
<td>inherited</td>
<td>Project Snapshots</td>
<td>Snapshots</td>
</tr>
<tr>
<td>space_available</td>
<td>read-only</td>
<td>Projects</td>
<td>Usage</td>
</tr>
<tr>
<td>space_data</td>
<td>read-only</td>
<td>Projects</td>
<td>Usage</td>
</tr>
</tbody>
</table>
Selecting a Cluster Pool using the CLI

<table>
<thead>
<tr>
<th>CLI Name</th>
<th>Type</th>
<th>BUI Name</th>
<th>BUI Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>space_snapshots</td>
<td>read-only</td>
<td>Projects</td>
<td>Usage</td>
</tr>
<tr>
<td>space_total</td>
<td>read-only</td>
<td>Projects</td>
<td>Usage</td>
</tr>
<tr>
<td>space_unused_res</td>
<td>read-only</td>
<td>Projects</td>
<td>Usage</td>
</tr>
<tr>
<td>space_unused_res_shares</td>
<td>read-only</td>
<td>Projects</td>
<td>Usage</td>
</tr>
<tr>
<td>vscan</td>
<td>inherited</td>
<td>Project - General</td>
<td>General</td>
</tr>
</tbody>
</table>

Shares Project General BUI Page

This section of the BUI controls overall settings for the project that are independent of any particular protocol and are not related to access control or snapshots. While the CLI groups all properties in a single list, this section describes the behavior of the properties in both contexts.

For information on how these properties map to the CLI, see “Projects CLI” on page 385.

- **Space Usage** - Space within a storage pool is shared between all shares. Filesystems can grow or shrink dynamically as needed, though it is also possible to enforce space restrictions on a per-share basis. For more information on pooled storage, see “Working with Shares”.

- **Quota** - Sets a maximum limit on the total amount of space consumed by all filesystems and LUNs within the project. For more information, see “Working with the Shares > Shares > General BUI Page” on page 309. Unlike filesystems, project quotas cannot exclude snapshots, and can only be enforced across all shares and their snapshots.

  Optionally, use the defaultuserquota property to set the default user quota and the defaultgroupquota property to set the default group quota for all filesystems within the project. These properties help you deploy default quotas for large numbers of users, including those belonging to groups. The defaults can be changed later for individual filesystems at the filesystem level.

**Note** - As part of capacity planning, remember to take into account default/user group quotas because the quotas could be exceeded if the source is larger than the destination. Also, shadow migration will fail if the target runs out of disk space.

- **Reservation** - Guarantees a minimum amount of space for use across all filesystems and LUNs within the project. For more information, see “Working with the Shares > Shares > General BUI Page” on page 309. Unlike filesystems, project reservation cannot exclude snapshots, and they can only be enforced across all shares and their snapshots.

- **Inherited Properties** - These are standard properties that can either be inherited by shares within the project. The behavior of these properties is identical to that at the shares level, and further documentation can be found in the shares section.
■ “Mountpoint” on page 309
■ “Read only” on page 309
■ “Update access time on read” on page 309
■ “Non-blocking mandatory locking” on page 309
■ “Data compression” on page 309
■ “Data deduplication” on page 309
■ “Checksum” on page 309
■ “Cache device usage” on page 309
■ “Database record size” on page 309
■ “Additional replication” on page 309
■ “Virus scan” on page 309

■ Custom Properties - Custom properties can be added as needed to attach user-defined tags to projects and shares. For more information, see “Working with Schemas” on page 391.

■ Filesystem Creation Defaults - These settings are used to fill in the default values when creating a filesystem. Changing them has no effect on existing filesystems. More information can be found in the appropriate shares section.
  ■ “User” on page 321
  ■ “Group” on page 321
  ■ “Permissions” on page 321

■ LUN Creation Defaults - These settings are used to fill in the default values when creating a LUN. Changing them has no effect on existing LUNs. More information can be found in the appropriate shares section.
  ■ “Volume size” on page 309
  ■ “Thin provisioned” on page 309
  ■ “Working with the Shares > Shares BUI Page” on page 295

Project Access

■ Access Control - This view provides control over inheritable properties that affect ACL behavior.

■ Inherited ACL Behavior - These properties behave the same way as at the share level. Changing the properties will change the corresponding behavior for any filesystems currently inheriting the properties.

For more information about ACLs, see “Working with the Shares > Shares > Access BUI Page” on page 321
Project Snapshots

Snapshots are read only copies of a filesystem at a given point of time. For more information on snapshots and how they work, see “Working with Snapshots” on page 286. Projects snapshots consist of snapshots of every filesystem and LUN in the project, all with identical names. Shares can delete the snapshots individually, and creating a snapshot with the same name as a project snapshot, while supported, can result in undefined behavior as the snapshot will be considered part of the project snapshot with the same name.

The following lists the project snapshot properties.

- **Scheduled Snapshot Label** - Filesystem snapshots can be accessed over data protocols at .zfs/snapshot in the root of the filesystem. This directory contains a list of all snapshots on the filesystem, and they can be accessed just like normal filesystem data (in read only mode). By default, the `.zfs` directory is not visible when listing directory contents, but can be accessed by explicitly looking it up. This prevents backup software from inadvertently backing up snapshots in addition to new data.

<table>
<thead>
<tr>
<th>BUI Value</th>
<th>CLI Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden</td>
<td>hidden</td>
<td>The .zfs directory is not visible when listing directory contents in the root of the filesystem. This is default.</td>
</tr>
<tr>
<td>Visible</td>
<td>visible</td>
<td>This .zfs directory appears like any other directory in the filesystem.</td>
</tr>
</tbody>
</table>

- **Scheduled Snapshot Label** - This optional property appends a user-defined label to each scheduled snapshot and is blank by default. The label can either be set for an individual share, or set for a project and inherited by its shares, but not both. Snapshot labels can help identify the project or share for which a snapshot was taken, for example "project1:share1" could indicate a scheduled snapshot taken on share1 within project1. Labels can be up to 35 alphanumeric characters and include special characters _ - . : 

  - Project level snapshots are administered in the same way as share level snapshots. For more information about snapshots, see “Shares:Snapshots” on page 328.
  - Project snapshots do not support rollback or clone operations. For more information about snapshots, see “Shares:Snapshots” on page 328.
  - To access the snapshots for a project, navigate to the project and run the `snapshots` command.

  ```
  clownfish:> shares select default
  clownfish:shares default> snapshots
  clownfish:shares default snapshots>
  ```

  - From this point, snapshots are administered in the same way as share level snapshots. For more information about snapshots, see “Shares:Snapshots” on page 328.
Working with Schemas

In addition to the standard built in properties, you can configure any number of additional properties that are available on all shares and projects. These properties are given basic types for validation purposes, and are inherited like most other standard properties. The values are never consumed by the software in any way, and exist solely for end-user consumption. The property schema is global to the system, across all pools, and is synchronized between cluster peers.

To work with schemas, see the following sections:
- “Configuring Schemas using the BUI” on page 391
- “Configuring a Schema using the BUI” on page 392
- “Configuring Schemas using the CLI” on page 392
- “Schemas”

Configuring Schemas using the BUI

To define custom properties, access the "Shares -> Schema" navigation item. The current schema is displayed as a list, and entries can be added or removed as needed. Each property has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>The CLI name for this property. This must contain only alphanumeric characters or the characters &quot;:_;&quot;.</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>The BUI name for this property. This can contain arbitrary characters and is used in the help section of the CLI.</td>
</tr>
<tr>
<td>TYPE</td>
<td>The property type, for validation purposes. This must be one of the types described below.</td>
</tr>
</tbody>
</table>

The valid types for properties are:

<table>
<thead>
<tr>
<th>BUI Type</th>
<th>CLI Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>String</td>
<td>Arbitrary string data. This is the equivalent of no validation.</td>
</tr>
</tbody>
</table>
Configuring a Schema using the BUI

1. Navigate to the Shares> Schema view.
2. Click the '+' icon to add a new property to the schema property list.
3. Enter the name of the property ("contact").
4. Enter a description of the property ("Owner Contact").
5. Choose a type for the new property ("Email Address").
6. Click Apply.
7. Navigate to an existing share or project.
8. Change the "Owner Contact" property under the "Custom Properties" section.

Configuring Schemas using the CLI

1. Navigate to the schema context (shares  schema).

<table>
<thead>
<tr>
<th>BUI Type</th>
<th>CLI Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>Integer</td>
<td>A positive or negative integer</td>
</tr>
<tr>
<td>Positive Integer</td>
<td>PositiveInteger</td>
<td>A positive integer</td>
</tr>
<tr>
<td>Boolean</td>
<td>Boolean</td>
<td>A true/false value. In the BUI this is presented as a checkbox, while in the CLI it must be one of the values &quot;true&quot; or &quot;false&quot;.</td>
</tr>
<tr>
<td>Email Address</td>
<td>EmailAddress</td>
<td>An email address. Only minimal syntactic validation is done.</td>
</tr>
<tr>
<td>Hostname or IP</td>
<td>Host</td>
<td>A valid DNS hostname or IP (v4 or v6) address.</td>
</tr>
</tbody>
</table>

Once defined, the properties are available under the general properties tab, using the description provided in the property table. Properties are identified by their CLI name, so renaming a property will have the effect of removing all existing settings on the system. A property that is removed and later renamed back to the original name will still refer to the previously set values. Changing the types of properties, while supported, may have undefined results on existing properties on the system. Existing properties will retain their current settings, even if they would be invalid given the new property type.
2. **Create a new property named “contact”** (create contact).

3. **Set the description for the property** (set description="Owner Contact").

4. **Set the type of the property** (set type=EmailAddress).

5. **Commit the changes** (commit).

6. **Navigate to an existing share or project.**

7. **Set the "custom:contact" property.**

---

**Example Schema**

The schema context can be found at Shares > Schema.

```
carp:> shares schema
```
```
carp:shares schema> show
```
```
Properties:
```
```
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>owner</td>
<td>EmailAddress</td>
<td>Owner Contact</td>
</tr>
</tbody>
</table>
```
```
```
Each property is a child of the schema context, using the name of the property as the token. To create a property, use the `create` command:
```
carp:shares schema> create department
```
```
carp:shares schema department (uncommitted)> get
type = String
description = department
```
```
carp:shares schema department (uncommitted)> set description="Department Code"
description = Department Code (uncommitted)
```
```
carp:shares schema department (uncommitted)> commit
carp:shares schema>
```
```
Within the context of a particular property, fields can be set using the standard CLI commands:
```
carp:shares schema> select owner
```
```
carp:shares schema owner> get
type = EmailAddress
description = Owner Contact
```
```
carp:shares schema owner> set description="Owner Contact Email"
description = Owner Contact Email (uncommitted)
```
```
carp:shares schema owner> commit
```
```
Once custom properties have been defined, they can be accessed like any other property under the name "custom:<property>":
Working with Data Encryption

```plaintext
carp:shares default> get
...  
custom:department = 123-45-6789
  custom:owner = 
...
carp:shares default> set custom:owner=bob@corp
  custom:owner = bob@corp (uncommitted)
carp:shares default> commit
```

**Working with Data Encryption**

**LICENSE NOTICE:** Encryption may be evaluated free of charge, but the feature requires that an independent license be purchased separately for use in production. Encryption is only available to license on the Oracle ZFS Storage ZS4-4 and Oracle ZFS Storage ZS3-4. After the evaluation period, this feature must either be licensed or deactivated. Oracle reserves the right to audit for licensing compliance at any time. For details, refer to the “Oracle Software License Agreement ("SLA") and Entitlement for Hardware Systems with Integrated Software Options.”

The Oracle ZFS Storage Appliance offers transparent data encryption for individual shares (filesystems and LUNs), and shares created inside of projects.

To configure and manage encryption, use these tasks:
- “Data Encryption Workflow” on page 395
- “Configuring LOCAL Keystore Encryption (BUI)” on page 395
- “Configuring LOCAL Keystore Encryption (CLI)” on page 397
- “Configuring OKM Keystore Encryption (BUI)” on page 398
- “Configuring OKM Keystore Encryption (CLI)” on page 399
- “Creating a Share (BUI)” on page 301
- “Creating an Encrypted Share (CLI)” on page 400
- “Changing a Project Encryption Key (BUI)” on page 401
- “Changing a Project Encryption Key (CLI)” on page 402
- “Changing a Share Encryption Key (BUI)” on page 403
- “Changing a Share Encryption Key (CLI)” on page 404
- “Deleting an Encryption Key (BUI)” on page 405
- “Deleting an Encryption Key (CLI)” on page 406
- “Restoring a LOCAL Key (CLI)” on page 407

To understand data encryption, use these topics:
- “Encryption Properties” on page 409
- “Managing Encryption Keys” on page 410
- “Performance Impact of Encryption” on page 412
- “Encryption Key Life Cycle” on page 413
Data Encryption Workflow

The following steps show the general procedure for configuring and using data encryption. For information about encryption properties, see “Encryption Properties” on page 409.

1. **Configure LOCAL keystore or Oracle Key Manager (OKM) keystore encryption.**
   For information about creating LOCAL or OKM keystores, see the following topics:
   - “Configuring LOCAL Keystore Encryption (BUI)” on page 395 or “Configuring LOCAL Keystore Encryption (CLI)” on page 397
   - “Configuring OKM Keystore Encryption (BUI)” on page 398 or “Configuring OKM Keystore Encryption (CLI)” on page 399

2. **Create LOCAL or OKM encryption keys.**
   For information about creating LOCAL or OKM keys, see the following topics:
   - “Configuring LOCAL Keystore Encryption (BUI)” on page 395 or “Configuring LOCAL Keystore Encryption (CLI)” on page 397
   - “Configuring OKM Keystore Encryption (BUI)” on page 398 or “Configuring OKM Keystore Encryption (CLI)” on page 399

3. **(Optional) Create a project using one of the LOCAL or OKM encryption keys.**
   For information about creating a project, see “Creating Projects” on page 384.

4. **Create a share in a project that uses an encryption key or create a share using one of the LOCAL or OKM encryption keys.**
   For information about creating a share, see “Creating a Share (BUI)” on page 301 or “Creating an Encrypted Share (CLI)” on page 400.

Configuring LOCAL Keystore Encryption (BUI)

To configure encryption using the LOCAL keystore, first set up the master passphrase and then create keys for assigning to encrypted shares. For information about encryption properties, see “Encryption Properties” on page 409.

To create a key, you provide the name to be used for assigning the key to projects or shares. You can choose to let the system generate the key value or you can supply a hex-encoded raw 256-bit key. Keys are stored in an encrypted form.
1. To configure the LOCAL keystore, go to the Shares > Encryption BUI page.

2. Click LOCAL.
   
   The LOCAL keystore information is displayed.

   ![Encryption Key Management Image]

3. To configure the master passphrase, type the passphrase supplied by your administrator and then retype it in the next box.

4. To save the master passphrase, click Apply or to discard your changes click Revert.

5. To create a key, click the Add item icon.
   
   The New Key dialog box is displayed.
6. Type a name for the key.

7. To use a hex-encoded raw 256-bit key, uncheck "Generate key automatically" and type the key.

8. To save the key, click Add, or to discard the changes click Cancel.

When you click Add, the new key appears in the list of keys with the creation date.

Configuring LOCAL Keystore Encryption (CLI)

This procedure assumes that encryption was not previously set up on the appliance. For information about encryption properties, see “Encryption Properties” on page 409.

1. To set up the master passphrase, use the following CLI commands:

brm7330-020:> shares encryption
brm7330-020:shares encryption> show
Children:
  okm => Manage encryption keys
  local => Manage encryption keys
To configure Okm Keystore Encryption (BUI)

To configure encryption using the Oracle Key Manager (OKM), first set up the Key Manager Server information, and then create keys for assigning to encrypted shares. For information about encryption properties, see “Encryption Properties” on page 409.

Note - If the appliance is clustered, do not use the "one time passphrase" setting when creating the OKM server agent otherwise registration on the other cluster node will fail and keys will not be available on failover.

To create a key, you provide the name to be used for assigning the key to projects or shares.

1. **To configure the OKM keystore, go to the Shares > Encryption BUI page.**

2. **Click OKM.**
The OKM keystore information is displayed.

3. To configure the server information, type the following information:
   - Key Manager Server
   - User Agent ID
   - Registration PIN

4. To save the server information, click Apply, or to discard the changes, click Cancel.

5. To create a key, click the Add item icon. The New Key dialog box is displayed.

6. Type a name for the key.

7. To save the key, click Add, or to discard the changes, click Cancel.
   When you click Add, the new key appears in the list of keys with the creation date.

▼ Configuring OKM Keystore Encryption (CLI)

To use the Oracle Key Manager (OKM) keystore, configure the following parameters:
   - agent_id
   - registration_pin (supplied by your OKM security officer)
   - server_addr
For information about encryption properties, see “Encryption Properties” on page 409.

**Note** - If the appliance is clustered, do not use the "one time passphrase" setting when creating the OKM server agent otherwise registration on the other cluster node will fail and keys will not be available on failover.

1. **To configure OKM keystore encryption, use the following CLI commands:**

   ```
   brm7330-020:> shares encryption
   brm7330-020:shares encryption> show
   Children:
       okm => Manage encryption keys
       local => Manage encryption keys
   
   brm7330-020:shares encryption> okm
   brm7330-020:shares encryption okm> show
   Properties:
       agent_id = ExternalClient041
       registration_pin = *********
       server_addr = 10.80.180.109
   
   
   Children:
       keys => Manage this Keystore's Keys
   ```

2. **To create an OKM key, use the following CLI commands:**

   ```
   brm7330-020:shares (pool-290-A) encryption okm keys>
   brm7330-020:shares (pool-290-A) encryption okm keys> create
   brm7330-020:shares (pool-290-A) encryption okm key-372 (uncommitted)> ls
   Properties:
       cipher = AES
       keyname = (unset)
   
   brm7330-020:shares (pool-290-A) encryption okm key-372 (uncommitted)> set
   keyname=anykey
   brm7330-020:shares (pool-290-A) encryption okm key-372 (uncommitted)> commit
   ```

![Creating an Encrypted Share (CLI)](image)

Creating an Encrypted Share (CLI)

Shares (filesystems and LUNs) can be encrypted individually or they can be encrypted at the project level because shares inherit project properties. The following example shows how to encrypt all shares within a project by encrypting the project itself. For information about encryption properties, see “Encryption Properties” on page 409.

1. **To create an encrypted project, use the following CLI commands:**

   ```
   brm7330-020:shares> project myproject
   ```
Changing a Project Encryption Key (BUI)

Changing a project encryption key changes the key for shares that inherit the key from the project. You can change the encryption key associated share at any time, even while it is in use by client systems. To change the key for a project, use the following procedure:

1. **To display the project you want to change, go to Shares > Projects.**

2. **To find the share you want, click Show All, Local, or Replica.**

3. **Move your cursor over the project you want to change, and click the Edit icon.**

4. **Click General. The project parameters are displayed.**

All shares created under this project are automatically encrypted with AES-128 CCM using the key named "MyFirstKey" from the LOCAL keystore.

2. **To create a filesystem in the new project and show that it inherited the encryption properties, use the following CLI commands:**

```bash
brm7330-020:shares> select myproject
brm7330-020:shares myproject> filesystem f1
brm7330-020:shares myproject/f1 (uncommitted)> commit
brm7330-020:shares myproject> select f1
brm7330-020:shares myproject/f1> get encryption keystore keyname keystatus
  encryption = aes-128-ccm (inherited)
  keystore = LOCAL (inherited)
  keyname = MyFirstKey (inherited)
  keystatus = available
brm7330-020:shares myproject/f1> done
```
5. To change the project encryption key, select Local or OKM and select the key you want to use.

6. To save the changes, click Apply or to discard your changes click Revert. When you click Apply, your changes are saved and the new key appears in the Encryption key area.

**Changing a Project Encryption Key (CLI)**

Changing a project encryption key changes the key for shares that inherit the key from the project. You can change the encryption key associated share at any time, even while it is in use by client systems. To change the key for a project, use the following procedure:

1. **To create an a new key, use the following CLI commands:**

   `brm7330-020:> shares encryption local keys create`
Changing a Share Encryption Key (BUI)

2. To change the key used for the project and all the associated shares, use the following CLI commands:

```
brm7330-020:> shares select myproject
brm7330-020:shares myproject> set keyname=MySecondKey
    keyname = MySecondKey (uncommitted)
brm7330-020:shares myproject> commit
```

3. To verify that a share has changed, navigate to the share and use the following CLI command:

```
shares test_project_encrypted/s1> get keyname
keyname = pascal1 (inherited)
brm7330-020:
```

Changing a Share Encryption Key (BUI)

You can change the encryption key associated share at any time, even while it is in use by client systems. To change a key for a share without changing the parent project, use the following procedure:

1. To display the properties for the share you want to change, go to Shares > Shares.

2. Select Filesystems or LUNS.

3. To find the share you want, click Show All, Local, or Replica.

4. Move your cursor over the share you want to change, and click the Edit icon 🖊️. The share properties are displayed.
5. If necessary, uncheck Inherit from project.

6. To change the encryption key, select Local or OKM and select the key you want to use.

7. To save the changes, click Apply or to discard your changes click Revert. When you click Apply, your changes are saved and the new key appears in the Encryption key area.

▼ Changing a Share Encryption Key (CLI)

You can change the encryption key associated share at any time, even while it is in use by client systems. To change a key for a share without changing the parent project, use the following procedure:

1. To change the key used for a share, use the following CLI commands:
Deleting an Encryption Key (BUI)

Deleting an encryption key is a fast and effective way to make large amounts of data inaccessible. Keys can be deleted even if they are in use. If the key is in use, a warning is given and confirmation is required. All shares or projects using that key are unshared and can no longer be accessed by clients.

1. **To delete an encryption key, go to the Shares > Encryption BUI page.**
2. **Select LOCAL or OKM.**
3. **Move your cursor over the key that you want to delete, click the Delete icon 🗑.**
   - The following alert is displayed:

   ```bash
   brm7330-020:> shares select myshare
   brm7330-020:shares myshare> set keyname=MyKey
       keyname = MyKey (uncommitted)
   brm7330-020:shares myshare> commit
   
   2. To verify that a share has changed, navigate to the share and use the following CLI command:

   shares test_project_encrypted/s1> get keyname
       keyname = pascal1 (inherited)brm7330-020:
   ```
4. **To delete the key, click OK or to keep the key click Cancel.**

   When a key is deleted, all of the data in all of the shares that use the key becomes inaccessible. This is equivalent to secure data destruction and is permanent and irrevocable, unless you have prepared for key restoration by backing up the key. For more information about key restoration, see “Restoring a LOCAL Key (CLI)” on page 407.

**Deleting an Encryption Key (CLI)**

Deleting an encryption key is a fast and effective way to make large amounts of data inaccessible. Keys can be deleted even if they are in use. If the key is in use, a warning is given
and confirmation is required. All shares or projects using that key are unshared and can no longer be accessed by clients.

1. **To delete a key, use the following CLI command:**

   brm7330-020:shares encryption local local_keys> destroy keyname=AKTEST_K1

   This key has the following dependent shares:
   
   - Pool2/local/BG1
   - Pool2/local/BG1/BG3
   - Pool2/local/BG1/fast1
   - Pool2/local/default/BG2

   Destroying this key will render the data inaccessible. Are you sure? (Y/N)

2. **To verify that a share is no longer accessible using that key, navigate to the share and use the following CLI commands:**

   brm7330-020:> shares select test_project select test_share1
   brm7330-020:shares test_project/test_share1> get encryption keystore keyname keystatus

   encryption = aes-128-ccm (inherited)
   keystore = LOCAL (inherited)
   keyname = AKTEST_K1 (inherited)
   keystatus = unavailable

   Errors:
   - key_unavailable

3. **To list dependents, use the following CLI commands:**

   brm7330-020:shares (pool-010) encryption local keys> select keyname=1
   brm7330-020:shares (pool-010) encryption local key-002> list

   Properties:
   - cipher = AES
   - keyname = 1

   brm7330-020:shares (pool-010) encryption local key-002> list dependents DEPENDENTS

---

**Restoring a LOCAL Key (CLI)**

Through LOCAL key restoration, an appliance administrator can create a LOCAL key with the same keyname and key value as a previously deleted key.

Key restoration gives the administrator the flexibility to render shares permanently or temporarily inaccessible by deleting the key. At a later point, a new LOCAL key with the same keyname and value can be re-created from a backup and associated with the shares, rendering them accessible again.
To enable LOCAL key restoration, back up key names and values before deleting keys from the LOCAL keystore.

For information about restoring keys stored in the OKM keystore, refer to the Oracle Key Manager documentation on the Oracle Technology Network (http://docs.oracle.com/).

▼ How to Back Up a LOCAL Key (CLI)

Use the following procedure to retrieve the information for a single LOCAL key and back up the key.

1. **Select the key:**
   
   hostname:shares encryption local keys> select keyname=Mykey

2. **Get the key value:**
   
   hostname:shares encryption local key-005> get key
   
   key = d6a5b01f0b93f2b1e7007a11d662d8092f2243c5d4c0cd34c264b15dd0b7739

3. **Back up the key name and value.**

▼ How to Delete a LOCAL Key (CLI)

- **To delete a LOCAL Key, use the destroy command:**
   
   hostname:shares encryption local keys> destroy keyname=Mykey
   
   This will destroy "key-005". Are you sure? (Y/N) Y

**Note** - If the key is in use, you will be given a list of shares using that key. If you want to keep any of these shares accessible, change those shares to use a different key before deleting the key they currently use.

▼ How to Restore a LOCAL Key (CLI)

Use the following procedure to restore a backed up LOCAL key.

**Note** - If the keyname of the key you are restoring is in use with a different key value for some existing shares, make sure to do a key change on those shares before restoring the original key.

1. **Create a key in the LOCAL keystore:**
   
   hostname:shares encryption local keys> create
2. Name the key based on the backup:

```bash
hostname:shares encryption local key-005 (uncommitted)> set keyname=Mykey
keyname = Mykey (uncommitted)
```

3. Set the key value based on the backup:

```bash
hostname:shares encryption local key-005 (uncommitted)> set
key=d6a5b01fbb93fcb19ef70a11d662d8092f243c5d4cc0cd34264b15dd0b7739
key = d6a5b01fbb93fcb19ef70a11d662d8092f243c5d4cc0cd34264b15dd0b7739 (uncommitted)
```

4. Save the key:

```bash
hostname:shares encryption local key-005 (uncommitted)> commit
```

You will be alerted if the new key being created is already referenced by existing shares. If this is acceptable, confirm the new key creation to complete restoration:

Existing shares reference the key Mykey from the LOCAL keystore. Are you sure? (Y/N) Y

## Encryption Properties

The following list shows the encryption properties available for managing keys, creating keys, and creating encrypted projects and shares.

- **LOCAL Key Management Properties**
  - **Master Passphrase** - The master passphrase is used to generate an AES key for encrypting the keys stored in the LOCAL keystore. The PKCS#5 PBKDF algorithm is used to generate the key and the key is randomly generated and managed by the system.

- **Local Key Creation Properties**
  - **Keyname** - Name to identify the key.
  - **Generate Key Automatically** - Automatically generate the key.
  - **Key** - Hex-encoded raw 256-bit key, stored in an encrypted form, if automatic key generation is not selected.

- **OKM Key Management Properties** (supplied by your OKM administrator)
  - **Key Manager Server** - IP address of your Oracle Key Manager (OKM) server.
  - **User Agent ID** - Agent ID.
  - **Registration PIN** - Registration PIN.

- **OKM Key Creation Properties**
  - **Keyname** - Name to identify the key.

- **Shares Encryption Properties**
  - **Encryption** - AES encryption type and key length (for more information, see “Understanding Encryption Key Values” on page 411).
How to Restore a LOCAL Key (CLI)

- **Inherit key** - Inherit the encryption key from the parent project.
- **Key** - Sets a specific LOCAL or OKM key and is used when the key is not inherited from the parent project.

**Project Encryption Properties**

- **Name** - Name to identify the project.
- **Encryption** - AES encryption type and key length (for more information, see “Understanding Encryption Key Values” on page 411.
- **Key** - Specific LOCAL or OKM key.

### Managing Encryption Keys

The appliance includes a built-in LOCAL keystore and the ability to connect to the Oracle Key Manager (OKM) system. Each encrypted project or share requires a wrapping key from either the LOCAL or OKM keystores. The data encryption keys are managed by the storage appliance and are stored persistently encrypted by the wrapping key from the LOCAL or OKM keystore.

OKM is a comprehensive key management system (KMS) that addresses the rapidly growing enterprise need for storage-based data encryption. Developed to comply with open standards, this feature provides the capacity, scalability, and interoperability to manage encryption keys centrally over widely distributed and heterogeneous storage infrastructures.

OKM meets the unique challenges of storage key management, including:

- **Long-term key retention** - OKM ensures that archive data is always available, and it securely retains encryption keys for the full data life cycle.
- **Interoperability** - OKM provides the interoperability needed to support a diverse range of storage devices attached to mainframe or open systems under a single storage key management service.
- **High availability** - With active N-node clustering, dynamic load balancing, and automated failover, OKM provides high availability, whether the appliances are sited together or distributed around the world.
- **High capacity** - OKM manages large numbers of storage devices and even more storage keys. A single clustered appliance can provide key management services for thousands of storage devices and millions of storage keys.
- **Flexible Key Configuration** - Per OKM cluster, keys can be generated automatically or created individually for a LOCAL or OKM keystore. Security administrators are responsible for providing the key names which, when combined with the keystore, associate a given wrapping key with a project or share.

**Note** - If the appliance is clustered, do not use the "one time passphrase" setting when creating the OKM server agent otherwise registration on the other cluster node will fail and keys will not be available on failover.
Maintaining Keys

Shares and projects that use OKM keys that are in a deactivated state remain accessible. To prevent an OKM key from being used, the OKM administrator must explicitly delete the key.

To ensure encrypted shares and projects are accessible, backup your appliance configurations and LOCAL keystore key values. If a key or keys become unavailable any shares or projects that use those keys become inaccessible. If a project's key is unavailable, new shares cannot be created in that project.

Keys can become unavailable in the following ways:
- Keys are deleted
- Roll back to a release that does not support encryption
- Roll back to a release where the keys are not configured
- Factory reset
- The OKM server is not available (OKM keys)

Understanding Encryption Key Values

The following table shows the BUI and CLI encryption key values and descriptions. It also indicates if the encryption type works with deduplication.

<table>
<thead>
<tr>
<th>TABLE 144 Encryption Key Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BUI Value</strong></td>
</tr>
<tr>
<td>Off</td>
</tr>
<tr>
<td>AES-128-CCM</td>
</tr>
<tr>
<td>AES-192-CCM</td>
</tr>
<tr>
<td>AES-256-CCM</td>
</tr>
<tr>
<td>AES-128-GCM</td>
</tr>
<tr>
<td>AES-192-GCM</td>
</tr>
<tr>
<td>AES-256-GCM</td>
</tr>
</tbody>
</table>
Understanding Encryption Errors

If an encryption key has expired, been deleted, or is incorrect, an error message will appear indicating the source of the problem. Use the following table to understand these errors:

<table>
<thead>
<tr>
<th>Error</th>
<th>Keystore</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key_incorrect</td>
<td>LOCAL or OKM</td>
<td>The value of the wrapping key assigned to this share does not match the value from when the share was first created or when the key last changed.</td>
</tr>
<tr>
<td>key_unavailable</td>
<td>LOCAL or OKM</td>
<td>The wrapping key assigned to this share has been deleted from the keystore. For an OKM key, this error could also mean that the remote key management server is currently unreachable.</td>
</tr>
<tr>
<td>key_unusable</td>
<td>OKM</td>
<td>The wrapping key assigned to this share has expired as per the policies configured in OKM.</td>
</tr>
</tbody>
</table>

To restore a LOCAL key that was previously backed up, see “Restoring a LOCAL Key (CLI)” on page 407

Performance Impact of Encryption

Using encryption with shares can have CPU performance impacts, as follows:

- The AES-128-CCM mode has the lowest CPU performance impact and is recommended for all workloads where there are no LOCAL security requirements.
- When encrypted data is read, it is stored decrypted and decompressed in DRAM. For read-dominant workloads that can be serviced read-dominant from the DRAM cache, the impact of decrypting the data is minimal.
- When SSD cache devices are used, data blocks evicted out of DRAM to the cache are compressed and encrypted and must be decrypted and decompressed when retrieved back into DRAM.
- For workloads that are write-dominant and use larger block sizes, especially 128 kilobytes and 1 megabyte, there can be a significant CPU impact resulting in lower throughput. This is particularly likely if the filesystem record size or LUN volume block size is larger than the application block size.
Encryption Key Life Cycle

The encryption key life cycle is flexible because you can change keys at any time without taking data services offline.

When a key is deleted from the keystore, all the shares that use it are unmounted and their data becomes inaccessible. Backing up keys in the OKM keystore should be performed using the OKM backup services. Backup of keys in the LOCAL keystore is included as part of the System Configuration Backup. For the LOCAL keystore, it is also possible to supply the key by value at creation time to allow it to be escrowed in an external system, which provides an alternative per-key backup/restore capability.

Backing up and Restoring Encrypted Data

When a share is restored using the ZFS restore function, the restored share inherits the encryption properties of the target project if the original share inherited its encryption properties from the source project.

To ensure encryption properties of an original share are maintained in a restored share, configure encryption on the original share instead of inheriting it from its project.

If you want to set encryption differently for an individual share within a project, manually configure encryption for the individual source share, instead of letting the share inherit its properties from the project. This ensures that all shares are backed up and restored with the desired encryption settings.

For more information about NDMP backup, see “Working with NDMP” on page 208. For information about replication, see “Working with Replication” on page 337.

Replicating an Encrypted Share

To replicate an encrypted share, both the source and target must support encryption and have:

- software release 2013.1.3.0 (or later)
- encryption wrapping keys used by the share

The replication will fail if you attempt to replicate an encrypted share and the target does not support encryption. If the wrapping key is not available on the source or target system, or the target software is earlier than 2013.1.3.0, an alert is raised. Review the alerts on both the source and target to determine the reason for the replication failure.

For more information on configuring replication, see “Working with Replication” on page 337.
Integrating Applications with the Oracle ZFS Storage Appliance

The Oracle ZFS Storage Appliance delivers a full suite of data protocols to communicate with a wide variety of application hosts. To improve application performance or more tightly integrate with your application environment, see the following:

- NAS Storage Documentation (http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/index.html)

For an example of how to achieve optimal results using standard protocols, see “FC Support with Symantec Dynamic Multi-Pathing and Storage Foundation” on page 449.

For some applications, installing software on the application host enhances interoperability. The following articles provide an overview of how software integration can provide a better experience for storage administrators. Comprehensive documentation is packaged with each download.

- “Appliance Plug-in for Oracle Solaris Cluster Geographic Edition” on page 446
- “Appliance Network File System Plug-in for Oracle Solaris Cluster” on page 446
- “Appliance Plug-in Provider for Volume Shadow Copy Service Software” on page 449
- “Appliance Plug-in for Oracle Enterprise Manager System Monitoring” on page 447
- “Appliance Plug-in for Oracle Virtual Machine Storage Connect” on page 448
- “Appliance Replication Adapter for VMware Site Recovery Manager” on page 450
- “Oracle Intelligent Storage Protocol” on page 445

With Oracle Snap Management Utility for Oracle Database, database administrators can ensure data protection, availability, and rapid testing by efficiently provisioning, cloning, backing up, and restoring an Oracle database for the Oracle ZFS Storage Appliance. For more information, see Oracle Snap Management Utility for Oracle Database (https://www.oracle.com/storage/nas/snap/index.html).

The Oracle ZFS Storage Appliance is engineered to seamlessly integrate with other Oracle products. The following sections describe how to configure the Oracle ZFS Storage Appliance...
as a target for Oracle database files and as a backup target for the Oracle Exadata Database Machine and the Oracle SuperCluster.

- “Oracle Exadata Database Machine Backup” on page 416
- “Oracle Exadata Database Machine Backup” on page 416
- “Oracle SuperCluster Backup” on page 424

Configuring the Oracle ZFS Storage Appliance for Oracle Database Clients

The Oracle ZFS Storage Appliance offers a number of unique features designed to integrate with Oracle database clients, including Hybrid Columnar Compression (HCC) and Oracle Intelligent Storage Protocol (OISP).

To enable these features, the SNMP service on the Oracle ZFS Storage Appliance must be configured to allow SNMP queries by database clients. The client uses this mechanism to identify a storage device as a ZFS Storage Appliance. For specifying the database client hostname or IP address as the trap destination, see “Configuring SNMP to Serve Appliance Status using the BUI” on page 252.

To verify the appliance SNMP service is configured properly, run the `snmpget(1)` command from the client system, replacing `<host>` with the name or IP address of the appliance.

```
-bash-4.1$ snmpget -v1 -c public <host> .1.3.6.1.4.1.42.2.225.1.4.2.0
SNMPv2-SMI::enterprises.42.2.225.1.4.2.0 = STRING: "Oracle ZFS Storage Appliance"
```

Oracle Exadata Database Machine Backup

When equipped with native QDR InfiniBand and 10Gb Ethernet connectivity options, the Oracle ZFS Storage Appliance is ideal for reliably backing up Oracle Exadata. The Oracle Exadata Backup Configuration Utility is provided for deployment using a command-line tool, or your appliance can be configured manually using the instructions in the following sections:

- “Configuring the Appliance for Exadata” on page 417
- “Configuring Exadata for the Appliance” on page 420

Comprehensive documentation is packaged with the utility, including instructions for how to execute a backup from the Oracle Exadata. Whether manually or using the utility, configuration of networking and storage pools on the appliance is required in addition to either approach.

For detailed information on using your ZFS Storage Appliance as a backup target for Oracle Exadata, see the Protecting Oracle Exadata with the Oracle ZFS Storage Appliance: Configuration Best Practices white paper on the NAS Storage Documentation page. Also available is an Oracle ZFS Storage ZS3-4 cluster, offered pre-racked with disk shelves as the
Oracle ZFS Storage ZS3-BA to minimize set-up complexity. Integrating this appliance with Oracle Exadata is identical to the process described above.

For detailed information, see the Protecting Oracle Exadata with the Oracle ZFS Storage Appliance: Configuration Best Practices white paper on the NAS Storage Documentation page.

Configuring the Appliance for Exadata

Use the following procedures to configure the appliance for Exadata.

- “Configuring the Appliance” on page 417
- “Configuring Appliance Pools” on page 418
- “Configuring Appliance Shares” on page 418
- “Configuring Oracle RMAN and the Oracle Database Instance” on page 419
- “Configuring Exadata for Appliance” on page 420

Configuring the Appliance Network

This section describes how to configure the IP network multipathing (IPMP) groups and how to configure routing in the ZFS Storage Appliance.

**Note** - If you used the Oracle Exadata Backup Configuration Utility, configure the network as described in this section. For details, review the Best Practices white paper.

For customers seeking additional IB connectivity, more IB HCAs can be installed and configured. For details, see the “Oracle ZFS Storage Appliance Installation Guide”.

The principles in this section can be applied to a 10Gb Ethernet implementation by applying the network configuration to the ixgbe interfaces instead of the ibp interfaces. The 10Gb Ethernet implementation may be configured as active/active IPMP. If the ZFS Storage Appliance is on a different subnet than the Oracle Exadata, it may be necessary to create static routes from the ZFS Storage Appliance to the Oracle Exadata. Consult with your network administrator for details.

1. **Ensure that the ZFS Storage Appliance is connected to the Oracle Exadata.**

2. **Configure ibp0, ibp1, ibp2, and ibp3 with address 0.0.0.0/8 (necessary for IPMP), connected mode, and partition key ffff.** To identify the partition key used by the Oracle Exadata system, run the following command as the root user:

   ```
   # cat /sys/class/net/ib0/pkey
   ```

3. **Configure the active/standby IPMP group over ibd0 and ibd3, with ibd0 active and ibd3 standby.**
4. **Configure the active/standby IPMP group over ibd1 and ibd2, with ibd2 active and ibd1 standby.**

5. **Enable adaptive routing to ensure traffic is load balanced appropriately when multiple IP addresses on the same subnet are owned by the same head. This occurs after a cluster failover.**

### Configuring Appliance Pools

This section describes design considerations to determine the most appropriate pool configuration for the ZFS Storage Appliance for Oracle Recovery Manager (RMAN) backup and restore operations based on data protection and performance requirements.

**Note** - If you used the Oracle Exadata Backup Configuration Utility, configure the pool as described in this section. For details, review the Best Practices white paper.

The system planner should consider pool protection based on the following guidelines:

- **Use parity-based protection for general-purpose and capacity-optimized systems:**
  - RAID-Z for protection from single-drive failure on systems subject to random workloads.
  - RAID-Z2 for protection from two-drive failure on systems with streaming workloads only.

- **Use mirroring for high-performance with incrementally applied backup.**

- **Configure pools based on performance requirements:**
  - Configure a single pool for management-optimized systems.
  - Configure two pools for performance-optimized systems. Two-pool systems should be configured by using half the drives from each tray.

- **Configure log device protection:**
  - Stripe log devices for RAID-Z and mirrored pool configurations.
  - Mirror log devices for RAID-Z2 pool configurations.

**Note** - If you used the Oracle Exadata Backup Configuration Utility, proceed to the next topic: “Configuring Exadata for The Appliance” on page 420.

### Configuring Appliance Shares

The default options for ZFS Storage Appliance shares provide a good starting point for general-purpose workloads. Appliance shares can be optimized for Oracle RMAN backup and restore operations as follows:

- Create a project to store all shares related to backup and recovery of a single database. For a two-pool implementation, create two projects; one for each pool.
Configure the shares supporting Oracle RMAN backup and restore workloads with the following values:

- Database record size \(\text{recordsize}\): 128kB
- Synchronous write bias \(\text{logbias}\): Throughput (for processing backup sets and image copies) or Latency (for incrementally applied backups)
- Cache device usage \(\text{secondary cache}\): None (for backup sets) or All (when supporting incrementally applied backups or database clone operations)
- Data compression \(\text{compression}\): Off for performance-optimized systems, LZJB or \text{gzip-2} for capacity-optimized systems
- Number of shares per pool: 1 for management-optimized systems, 2 or 4 for performance-optimized systems

Additional share configuration options, such as higher-level \text{gzip} compression or replication, can be applied to shares used to support Oracle Exadata backup and restore, as customer requirements mandate.

Customers implementing additional ZFS Storage Appliance data services should consider implementation-specific testing to verify the implications of deviations from the practices described earlier.

### Configuring Oracle RMAN and the Oracle Database Instance

Oracle RMAN is an essential component for protecting the content of Oracle Exadata. Oracle RMAN can be used to create backup sets, image copies, and incrementally updated backups of Oracle Exadata content on ZFS Storage Appliances. To optimize performance of Oracle RMAN backups from Oracle Exadata to a ZFS Storage Appliance, the database administrator should apply the following best practices:

- Load balance Oracle RMAN channels evenly across the nodes of the database machine.
- Load balance Oracle RMAN channels evenly across ZFS Storage Appliance shares and controllers.

To optimize buffering of the Oracle RMAN channel to the ZFS Storage Appliance, you can tune the values of several hidden instance parameters. For Oracle Database 11g Release 2, the following parameters can be tuned:

- For backup and restore set:
  - \_backup_disk_bufcnt=64
  - \_backup_disk_bufsz=1048576
- For image copy backup and restore:
  - \_backup_file_bufcnt=64
  - \_backup_file_bufsz=1048576

For additional information about tuning these parameters and tuning equivalent parameters for earlier versions of the Oracle Database software, see Article ID 1072545.1: RMAN
Performance Tuning Using Buffer Memory Parameters) at My Oracle Support (https://support.oracle.com).

Oracle Direct NFS (dNFS) is a high-performance NFS client that delivers exceptional performance for Oracle RMAN backup and restore operations. dNFS should be configured for customers seeking maximum throughput for backup and restore operations.

Configuring Exadata for Appliance

If you use the Oracle Exadata Backup Configuration Utility, all steps except for step 4 and step 7 have already been performed for you.

- “Configuring the Appliance for Exadata” on page 417
- “Configuring the Directory Structure for the Shares” on page 420
- “Updating the /etc/fstab File” on page 420
- “Creating an init.d Service” on page 421
- “Updating oranfstab to Access Appliance Exports” on page 422
- “Mounting the Shares on the Host” on page 423
- “Setting the Ownership of the Mounted Shares” on page 423

▼ Configuring the Directory Structure for the Shares

- Set up mount points for the shares on the host as shown:

  ```bash
  mkdir -p /zfssa/dbname/backup1
  mkdir -p /zfssa/dbname/backup2
  mkdir -p /zfssa/dbname/backup3
  mkdir -p /zfssa/dbname/backup4
  ```

▼ Updating the /etc/fstab File

The UNIX new-line escape character (\) indicates a single line of code has been wrapped to a second line in the listing below. When entering a wrapped line into `fstab`, remove the \ character and combine the two line segments, separated by a space, into a single line.

1. To update the /etc/fstab file, use one of the following options.

   For a one-pool configuration:

   ```bash
   192.168.36.200:/export/dbname/backup1 /zfssa/dbname/backup1 nfs \\
   rw, bg, hard, nointr, rsize=1048576, wsize=1048576, tcp, nfsvers=3, timeo=600 0 0
   192.168.36.200:/export/dbname/backup2 /zfssa/dbname/backup2 nfs \\
   rw, bg, hard, nointr, rsize=1048576, wsize=1048576, tcp, nfsvers=3, timeo=600 0 0
   ```
Creating an init.d Service

1. Create an init.d service using the appropriate following option.

```
#!/bin/sh
#
# zfssa_dbname: Mount ZFSSA project dbname for database dbname
#
# chkconfig: 345 61 19
# description: mounts ZFS Storage Appliance shares
#
start()
{
    mount /zfssa/dbname/backup1
    mount /zfssa/dbname/backup2
    mount /zfssa/dbname/backup3
    mount /zfssa/dbname/backup4
    echo "Starting $prog:"
}

stop()
{
    umount /zfssa/dbname/backup1
    umount /zfssa/dbname/backup2
    umount /zfssa/dbname/backup3
    umount /zfssa/dbname/backup4
    echo "Stopping $prog:"
}

case "$1" in
    start)
        start
        ;;
    stop)
```

2. For a two-pool configuration:

```
192.168.36.200:/export/dbname/backup1 /zfssa/dbname/backup1 nfs <br/>
   rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \br/>   3,timeo=600 0 0
192.168.36.200:/export/dbname/backup2 /zfssa/dbname/backup2 nfs <br/>
   rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \br/>   3,timeo=600 0 0
192.168.36.200:/export/dbname/backup3 /zfssa/dbname/backup3 nfs <br/>
   rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \br/>   3,timeo=600 0 0
192.168.36.200:/export/dbname/backup4 /zfssa/dbname/backup4 nfs <br/>
   rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \br/>   3,timeo=600 0 0

▼ Creating an init.d Service

1. Create an init.d service using the appropriate following option.

```
#!/bin/sh
#
# zfssa_dbname: Mount ZFSSA project dbname for database dbname
#
# chkconfig: 345 61 19
# description: mounts ZFS Storage Appliance shares
#
start()
{
    mount /zfssa/dbname/backup1
    mount /zfssa/dbname/backup2
    mount /zfssa/dbname/backup3
    mount /zfssa/dbname/backup4
    echo "Starting $prog:"
}

stop()
{
    umount /zfssa/dbname/backup1
    umount /zfssa/dbname/backup2
    umount /zfssa/dbname/backup3
    umount /zfssa/dbname/backup4
    echo "Stopping $prog:"
}

case "$1" in
    start)
        start
        ;;
    stop)
```
Updating oranfstab to Access Appliance Exports

stop
;
restart
stop
start
;
status
mount
;
*
)
echo "Usage: $0 {start|stop|restart|status}"
exit 1
esac

2. (Optional) Enable the *init.d* service for start-on-boot by entering:

# chkconfig zfssa_dbname on

3. (Optional) Start and stop the service manually using the service commands:

# service zfssa_dbname start
# service zfssa_dbname stop

▼ Updating oranfstab to Access Appliance Exports

If you used the Oracle Exadata Backup Configuration Utility, you may optionally perform this procedure.

1. To update the *oranfstab* file to access ZFS Storage Appliance exports, use the appropriate following option. For a one-pool configuration:

server: 192.168.36.200
path: 192.168.36.200
export: /export/dbname/backup1 mount: /zfssa/dbname/backup1
export: /export/dbname/backup2 mount: /zfssa/dbname/backup2
export: /export/dbname/backup3 mount: /zfssa/dbname/backup3
export: /export/dbname/backup4 mount: /zfssa/dbname/backup4

2. For a two-pool configuration:

server: 192.168.36.200
path: 192.168.36.200
export: /export/dbname/backup1 mount: /zfssa/dbname-2pool/backup1
export: /export/dbname/backup2 mount: /zfssa/dbname-2pool/backup2
export: /export/dbname/backup3 mount: /zfssa/dbname-2pool/backup3
d server: 192.168.36.201
path: 192.168.36.201
export: /export/dbname/backup2 mount: /zfssa/dbname-2pool/backup2
export: /export/dbname/backup4 mount: /zfssa/dbname-2pool/backup4
Mounting the Shares on the Host

1. To mount the shares on the host, enter one of the following two options:

   ```
   # service mount_dbname start
   ```

2. or

   ```
   # dcli -l root -g /home/oracle/dbs_group service mount_dbname start
   ```

Setting the Ownership of the Mounted Shares

Change the permission settings of the mounted shares to match the permission settings of `ORACLE_HOME`. In this example, the user and group ownerships are set to `oracle:dba`.

If you used the Oracle Exadata Backup Configuration Utility, you may optionally perform step 2; step 1 has already been performed for you.

1. Enter one of the following two options:

   ```
   <br/>
   # chown oracle:dba /zfssa/dbname/ *
   ```

   ```
   or<br/>
   # dcli -l root -g /home/oracle/dbs_group chown oracle:dba/zfssa/dbname/*
   ```

2. Restart the Oracle Database instance to pick up the changes that were made to the `orafstab` file using one of the following options:

3. Restart one instance at a time (rolling upgrade).

**Example 24**

Restarting One Instance at a Time

```bash
$ srvctl stop instance -d dbname -i dbname1  
$ srvctl stop instance -d dbname -i dbname1  
$ srvctl start instance -d dbname -i dbname1  
$ srvctl stop instance -d dbname -i dbname2  
$ srvctl start instance -d dbname -i dbname2  
$ srvctl stop instance -d dbname -i dbname3  
$ srvctl start instance -d dbname -i dbname3  
$ srvctl stop instance -d dbname -i dbname4  
$ srvctl start instance -d dbname -i dbname4  
$ srvctl stop instance -d dbname -i dbname5  
$ srvctl start instance -d dbname -i dbname5  
$ srvctl stop instance -d dbname -i dbname6  
$ srvctl start instance -d dbname -i dbname6  
$ srvctl stop instance -d dbname -i dbname7  
$ srvctl start instance -d dbname -i dbname7  
$ srvctl stop instance -d dbname -i dbname8  
$ srvctl start instance -d dbname -i dbname8
```
Oracle SuperCluster Backup

Restart the entire database, for example:

```bash
$ srvctl stop database -d dbname
$ srvctl start database -d dbname
```

Oracle SuperCluster Backup

When equipped with native QDR InfiniBand and 10Gb Ethernet connectivity options, the ZFS Storage Appliance is ideal for reliably backing up the Oracle SPARC SuperCluster. Use the instructions in the following sections to configure your system:

- “Configuring the Appliance for SuperCluster Backup” on page 424
- “Configuring Oracle SPARC SuperCluster for Appliance Backup” on page 442

For detailed information on using your ZFS Storage Appliance as a backup target for Oracle SPARC SuperCluster, see the Configuring a Sun ZFS Backup Appliance with Oracle SPARC SuperCluster white paper on the NAS Storage Documentation page. Also available is an Oracle ZFS Storage ZS3-4 cluster, offered pre-racked with disk shelves as the Oracle ZFS Storage ZS3-BA to minimize set-up complexity. Integrating this appliance with Oracle SPARC SuperCluster is identical to the process described above.

Configuring the Appliance for SuperCluster Backup

This procedure provides general steps for configuring a ZFS Storage Appliance for backup use with the Oracle SPARC SuperCluster. For detailed information, see the white paper Configuring a Sun ZFS Backup Appliance with Oracle SPARC SuperCluster on the NAS Storage Documentation page.

- “Configuring the ZFS Storage Appliance InfiniBand Datalinks” on page 425
- “Configuring the Oracle SPARC SuperCluster InfiniBand Switches to Add the ZFS Storage Appliance” on page 425
- “Configuring ZFS Storage Appliance Networking for Single IP Connection” on page 428
- “Configuring ZFS Storage Appliance Networking for an Active-Active Configuration” on page 429
- “Configuring the ZFS Storage Appliance Storage Pool” on page 431
- “Configuring the ZFS Storage Appliance Shares” on page 431
- “Configuring the ZFS Storage Appliance DTrace Analytics” on page 432
- “Configuring the Client NFS Mount” on page 433
- “Tuning the Solaris 11 Network and Kernel” on page 433
- “Configuring Oracle Direct NFS (dNFS)” on page 433
Configuring the ZFS Storage Appliance InfiniBand Datalinks

Follow the steps in this section to configure each ZFS Storage Appliance InfiniBand connection. The eight GUIDs for the InfiniBand HBA ports that are recorded during this procedure are used to configure the Oracle SPARC SuperCluster InfiniBand switches in the next procedure.

1. Connect the ZFS Storage Appliance to the Oracle SPARC SuperCluster as described in the white paper Configuring a Sun ZFS Backup Appliance with Oracle SPARC SuperCluster on the NAS Storage Documentation page.

2. Log on to the Browser User Interface (BUI) of Head 1 and navigate to Configuration > Network.

3. Click the plus icon next to Datalinks. The Network Datalink dialogue box opens.

4. Complete the dialogue box as follows:
   - Check the IB Partition box.
   - Enter a meaningful name for the datalink name.
   - Set the Partition Key to 8503.
   - Select Connected Mode for the Link Mode.
   - Do not check the LACP Aggregation box.
   - Select Partition Device ibp0.
   - Record the GUID number (for example, 21280001ef43bb) and click Apply.

5. Repeat steps 3 and 4 for each remaining InfiniBand interface (ibp1, ibp2, and ibp3).

6. Repeat steps 2 through 5 for Head 2.

Configuring the Oracle SPARC SuperCluster InfiniBand Switches to Add the ZFS Storage Appliance

In this procedure, the GUIDs of the ZFS Storage Appliance InfiniBand HBA ports are added to the existing Oracle SPARC SuperCluster InfiniBand configuration. By adding these ports and using a partition key of 8503, communication between the two devices can occur.
1. Log on to the Oracle SPARC SuperCluster InfiniBand spine switch as root. By default, the spine switch is given a hostname of `<sscid>sw-ib1`, where `<sscid>` is the prefix name given to the entire Oracle SPARC SuperCluster system. In the following example, the `<sscid>` is aiessc.

   ```
   login as: root
   root@aiesscsw-ib1's password:
   Last login: Tue Sep 25 08:19:01 2013 from dhcp-brm-bl5-204-3e
   east-10-135-75-254.usdhcp.oraclecorp.com
   ```

2. Enter the command `enablesm` to verify that the switch is running Subnet Manager (or this command will start Subnet Manager).

   ```
   [root@aiesscsw-ib1 ~]# enablesm
   opensm (pid 15906) is already running...
   Starting partitiond daemon
   /usr/local/util/partitiond is already running
   (You may also perform a 'restart' if wanted)
   ```

3. Enter the command `getmaster` to verify that this is the master switch of the configuration. If the master switch is not running on the spine switch, log out and log in to the designated master switch for the remainder of this procedure.

   ```
   [root@aiesscsw-ib1 ~]# getmaster
   Local SM enabled and running
   20130913 10:16:51 Master SubnetManager on sm lid 13 sm guid
   0x2128e8ac27a0a0 : SUN DCS 36P QDR aiesscsw-ib1.us.oracle.com
   [root@aiesscsw-ib1 ~]#
   ```

4. Back up the switch configuration according the documented backup procedures (http://docs.oracle.com/cd/E26698_01/index.html).

5. Enter the command `smpartition list active` to verify that partition key 0x0503 is assigned to partition name "sto" (sto = 0x0503). The partition key was set to 8503 on the ZFS Storage Appliance datalinks, but the InfiniBand switch reports 0503. This is intentional because the InfiniBand protocol reserves the most significant bit (0x8000) of the hexadecimal partition key (pkey) for its own use. Therefore, pkeys 0x8000 and 0x0503 are the same.

   ```
   [root@aiesscsw-ib1 ~]# smpartition list active
   # Sun DCS IB partition config file
   # This file is generated, do not edit
   ! version_number : 11
   Default=0x7fff, ipoib : ALL_CAS=full, ALL_SWITCHES=full, SELF=full;
   SUN_DCS=0x0001, ipoib : ALL_SWITCHES=full;
   ```
6. **Add the ZFS Storage Appliance to the InfiniBand configuration:**

7. **Enter the command** `smpartition start` **to start a reconfiguration session.**

   ```bash
   # smpartition start
   [root@aiesscsw-ib1 ~]# smpartition start
   ``

8. **Enter the command** `smpartition add` **to add the eight new GUIDs to the configuration.**

   ```bash
   # smpartition add -n sto -port <GUID1> <GUID2> <GUID3> ... <GUID8>
   [root@aiesscsw-ib1 ~]# smpartition add -n sto -port
   21280001ef43bb 21280001ef43bc 21280001cf90bf 21280001cf90c0
   21280001ef43f7 21280001ef43f8 21280001ef43b7 21280001ef43b0
   ``

9. **Enter the command** `smpartition list modified` **to verify the new GUIDs have been added correctly.**

   ```bash
   # smpartition list modified
   [root@aiesscsw-ib1 ~]# smpartition list modified
   # Sun DCS IB partition config file
   # This file is generated, do not edit
   #! version_number : 11
   Default=0x7fff, ipoib : ALL_CAS=full, ALL_SWITCHES=full, SELF=full;
   SUN_DCS=0x0001, ipoib : ALL_SWITCHES=full;
   ic1s10 = 0x0501,ipoib,defmember=full:
   0x002128001ef30f7,
   0x002128001ef30f8,
   0x002128001ef30b7,
   0x002128001ef314b;
   ic2s10 = 0x0502,ipoib,defmember=full:
   0x002128001ef30f8,
   0x002128001ef33c0,
   0x002128001ef30b8,
   0x002128001ef314c;
   sto = 0x0503,ipoib,defmember=full:
   0x002128001ef43f8,
   0x002128001ef43b7,
   0x002128001cf90c0,
   0x002128001ef43bb,
   ...more...
   ```
10. **Enter the command** `smpartition commit` **to apply the new configuration and propagate configuration changes to all InfiniBand switches in the configuration.**

    # smpartition commit
    [root@aiesscsw-ib1 ~]# smpartition commit
    [root@aiesscsw-ib1 ~]#

11. **Log off the InfiniBand switch.**

12. **Back up the InfiniBand configuration according to the documented backup procedures** Sun Datacenter InfiniBand Switch Documentation (http://docs.oracle.com/cd/E26698_01/index.html).

▼ **Configuring ZFS Storage Appliance Networking for Single IP Connection**

This configuration is only for an Oracle SPARC SuperCluster T5 with no external leaf switches. For best failover and performance, use the Active-Active Configuration (next section) for all other configurations.
Configure the ZFS Storage Appliance InfiniBand ports for network connectivity and simple cluster failover by using the following procedure to configure Port 1 with the desired IP address.

1. **Log on to the BUI of Head 1 and navigate to Configuration > Network.**

2. **Click the plus icon next to Interfaces. The Network Interface dialogue box opens.**

3. **Complete the dialogue box as follows:**
   - Enter a meaningful name for the network interface.
   - Verify that **Enable Interface** is checked.
   - Verify that **Allow Administration** is checked.
   - Verify that **Use IPv4 Protocol** is checked.
   - Verify that the **Configure with** menu selection is **Static Address List**.
   - In the box below that, enter the desired IP address with the appropriate netmask.
   - Verify that **Use IPv6 Protocol** is not checked.
   - Select the datalink for ibp0 and click **Apply**.

4. **Repeat steps 1 through 3 on Head 2 using ibp2 as the datalink.**

**Configuring ZFS Storage Appliance Networking for an Active-Active Configuration**

Configure the InfiniBand ports on the ZFS Storage Appliance for IP multipathing. Four IP addresses, on the private storage subnet, are needed for each ZFS Storage Appliance head (therefore, eight addresses total) because the interfaces will run in an active-active configuration.

1. **Configure the IPMP interface on Head 1.**
   - Log on to the BUI of Head 1 and navigate to Configuration > Network.
   - Click the plus icon next to Interfaces. The Network Interface dialogue box opens.
   - Complete the dialogue box as follows:
     - Enter a meaningful name for the IPMP network interface.
     - Verify that **Enable Interface** is checked.
     - Verify that **Allow Administration** is checked.
     - Verify that **Use IPv4 Protocol** is checked.
     - Verify that the **Configure with** menu selection is **Static Address List**.
     - Click the plus sign next to the empty box three times, so that four empty boxes are displayed.
In each empty box, enter one of the IP addresses reserved for the InfiniBand connections with its respective /24 netmask designation. As a best practice, do not use consecutive IP addresses from the block, but rather every other one (for example, all odd or all even).

- Verify that Use IPv6 Protocol is not checked.
- Check the IP MultiPathing Group box.
- Check the boxes next to the interfaces corresponding to datalinks ibp0 and ibp3.
- Verify that each of the two interfaces are set to Active and click Apply.

From Configuration > Network, click Routing.

- Click on the Multihoming model corresponding to Adaptive.

2. Configure the IPMP interface on Head 2.
   - Log on to the BUI of Head 2 and navigate to Configuration > Network.
   - Click the plus icon next to Interfaces. The Network Interface dialogue box opens.
   - Complete the dialogue box as follows:
     - Enter a meaningful name for the IPMP network interface.
     - Verify that Enable Interface is checked.
     - Verify that Allow Administration is checked.
     - Verify that Use IPv4 Protocol is checked.
     - Verify that the Configure with menu selection is Static Address List.
     - Click the plus sign next to the empty box three times, so that four empty boxes are displayed.
     - In each empty box, enter one of the remaining four IP addresses reserved for the InfiniBand connections with its respective /24 netmask designation. These should be the ones not used on Head 1.
     - Verify that Use IPv6 Protocol is not checked.
     - Check the IP MultiPathing Group box.
     - Check the boxes next to the interfaces corresponding to datalinks ibp1 and ibp2.
     - Verify that each of the two interfaces are set to Active and click Apply.

From Configuration > Network, click Routing.

- Click on the Multihoming model corresponding to Adaptive.

3. Verify connectivity with the Oracle SPARC SuperCluster nodes. Verify that each node can ping each of the eight addresses used in the IPMP groups on the ZFS Storage Appliance. Add these IP addresses to the /etc/inet/hosts table of each node.
Configuring the ZFS Storage Appliance Storage Pool

Pool configuration assigns physical disk drive resources to logical storage pools for backup data storage. To maximize system throughput, configure two equally sized storage pools by assigning half of the physical drives in each drive tray to each storage pool.

The ZFS Storage Appliance management software presents a warning message about efficiency when two pools with the same RAID protection profile are configured. This message can be safely ignored when configuring for a high-performance Oracle RMAN backup solution.

Configuring the ZFS Storage Appliance Shares

Share configuration is the process of setting up and running NFS mount points for client access. Two projects should be created for the Oracle SPARC SuperCluster configuration: one project per pool. A project is an entity that provides a higher level management interface point for a collection of shares. To optimize share management, update the default mount point for shares contained in the project to reference the database name, such as /export/dbname. For a performance-optimized system, create four shares for each project in each pool, for a total of eight shares (four for each head). To configure a project, perform the following:

1. Log on to the BUI of Head 1 and navigate to Shares > Projects.
2. Click the plus icon next to Projects, enter a meaningful name for the project, and click Apply. Since a similar project will be created on the other head, uniquely name the project for Head 1, such as H1-mydb.
3. Click the pencil icon next to the new project name to edit the project.
4. Click General and complete the properties as follows:
   - Change the Mountpoint to include the database name (for example, /export/H1-mydb).
   - Change Synchronous write bias from Latency to Throughput and click Apply.
5. Click Protocols and add an NFS exception as follows:
   - Click the plus icon next to NFS Exceptions.
   - Change Type to Network.
   - Enter the subnet and netmask (for example, /24) of the InfiniBand network.
   - Change Access Mode to Read/Write.
   - Verify that Charset is set to default.
   - Check the Root Access box and click Apply.
6. Next to General, click Shares.
7. Create four filesystems for Head 1 and uniquely name them so they will be different from the names for Head 2. To interleave the backup streams to distribute the data across the two heads and, thereby, provide better performance, use odd-numbered names for Head 1, such as backup1, backup3, backup5, and backup7; and use even-numbered names for Head 2, such as backup2, backup4, backup6, and backup8. To create the filesystems, click the plus icon next to Filesystems, enter the name of the filesystem (backup1), and click Apply. Repeat this step to create the remaining three filesystems (backup3, backup5, and backup7).

8. Repeat steps 1 through 7 for Head 2. Remember to use a unique project name (for example, H2-mydb), and specify even-numbered backup IDs (backup2, backup4, backup6, and backup8) for the filesystem names.

Configuring the ZFS Storage Appliance DTrace Analytics

The ZFS Storage Appliance includes a comprehensive performance analysis tool called DTrace Analytics. DTrace Analytics is a framework that monitors important subsystem performance accounting statistics. A subset of the available accounting statistics should be monitored to provide comprehensive data on the effectiveness and performance of Oracle RMAN backup and restore workloads.

The following Analytics are available when advanced analytics are configured on the ZFS Storage Appliance (Configuration > Preferences > Enable Advanced Analytics):
- CPU: Percent utilization broken down by CPU mode
- Disk: Average number of I/O operations broken down by state of operation
- Disk: I/O bytes per second broken down by type of operation
- Disk: I/O operations per second broken down by latency
- Disk: Disks with utilization of at least 95 percent broken down by disk
- Network: Interface bytes per second broken down by direction
- Network: Interface bytes per second broken down by interface
- Protocol: NFSv3 operations per second broken down by size
- Protocol: NFSv3 operations per second broken down by type of operation
- Protocol: NFSv3 operations per second of type read broken down by latency
- Protocol: NFSv3 operations per second of type write broken down by latency
- Protocol: NFSv3 operations per second of type read broken down by size
- Protocol: NFSv3 operations per second of type write broken down by size

Implementing these accounting statistics helps end-users gain a quantitative understanding of the instantaneous and historical resource consumption and quality of service (QoS) for their specific implementation.
Configuring the Client NFS Mount

When configuring the ZFS Storage Appliance, any server that accesses the appliance, including Oracle SPARC SuperCluster nodes, is considered a client. Configuring the client NFS mount includes creating the target directory structure for access to the ZFS Storage Appliance as well as the specific NFS mount options necessary for optimal system performance. Mount options for Solaris clients are:

\[ \text{rw, bg, hard, nointr, rsize=1048576, wsize=1048576, proto=tcp, vers=3, forcedirectio} \]

The mount points of the directories created on the ZFS Storage Appliance should be created on each of the Oracle SPARC SuperCluster nodes and added to their \(/etc/inet/hosts\) table.

Tuning the Solaris 11 Network and Kernel

1. The following entries should be added to the \(/etc/system\) file of each Oracle SPARC SuperCluster node:

   \[
   \text{set rpcmod:clnt_max_conns = 8} \\
   \text{set nfs:nfs3_bsize = 131072}
   \]

2. Additionally, the following commands need to be run on each Oracle SPARC SuperCluster node every time it is rebooted:

   \[
   \text{/usr/sbin/ndd -set /dev/tcp tcp_max_buf 2097152} \\
   \text{/usr/sbin/ndd -set /dev/tcp tcp_xmit_hiwat 1048576} \\
   \text{/usr/sbin/ndd -set /dev/tcp tcp_recv_hiwat 1048576}
   \]

3. Additional tuning might be necessary to achieve optimal performance. Refer to Oracle SPARC SuperCluster Tunables document 1474401.1, available at My Oracle Support (http://support.oracle.com), for the latest information. Also, the January 2013 QFSDP release added a "ssctuner" tool that automatically sets tunables. Refer to the Oracle SPARC SuperCluster release notes for additional information.

Configuring Oracle Direct NFS (dNFS)

On each Oracle SPARC SuperCluster node, configure dNFS as follows:

1. Shut down the running instance of the Oracle Database software.

2. Change directory to \$/ORACLE_HOME/rdbms/lib.
3. **Enable dNFS:**

   ```bash
   make -f $ORACLE_HOME/rdbms/lib/ins_rdbms.mk dnfs_on
   ```

4. **Update the `oranfstab` file (located in `/ORACLE_HOME/dbs`) with the server, path, and export names specific to the configuration, where:**

   - The server parameter refers to the local name of the ZFS Storage Appliance head on the InfiniBand network.
   - The path parameters should reflect the address(es) for that head specified during configuration.
   - The export parameters should reflect the mount points similar to the entries created in `/etc/vfstab`. The entries should look similar to the following.

5. **For single IP configuration (only Oracle SPARC SuperCluster T5 without external leaf switches):**

   ```
   server: aie-zba-h1-stor
   path: 192.168.30.100
   export: /export/test1/backup1 mount: /zba/test1/backup1
   export: /export/test1/backup3 mount: /zba/test1/backup3
   export: /export/test1/backup5 mount: /zba/test1/backup5
   export: /export/test1/backup7 mount: /zba/test1/backup7
   server: aie-zba-h2-stor
   path: 192.168.30.101
   export: /export/test1/backup2 mount: /zba/test1/backup2
   export: /export/test1/backup4 mount: /zba/test1/backup4
   export: /export/test1/backup6 mount: /zba/test1/backup6
   export: /export/test1/backup8 mount: /zba/test1/backup8
   ```

6. **For IPMP Group configuration (all others):**

   ```
   server: aie-zba-h1-stor
   path: 192.168.30.100
   path: 192.168.30.102
   path: 192.168.30.104
   path: 192.168.30.106
   export: /export/test1/backup1 mount: /zba/test1/backup1
   export: /export/test1/backup3 mount: /zba/test1/backup3
   export: /export/test1/backup5 mount: /zba/test1/backup5
   export: /export/test1/backup7 mount: /zba/test1/backup7
   server: aie-zba-h2-stor
   path: 192.168.30.101
   path: 192.168.30.103
   path: 192.168.30.105
   path: 192.168.30.107
   export: /export/test1/backup2 mount: /zba/test1/backup2
   export: /export/test1/backup4 mount: /zba/test1/backup4
   export: /export/test1/backup6 mount: /zba/test1/backup6
   ```
7. **Restart the Oracle Database software instance.**

**Tuning the Oracle Database Instance for Oracle RMAN Backup and Restore**

Optimizing high-bandwidth backup and restore operations using Oracle RMAN and the ZFS Storage Appliance requires adjusting the instance parameters that control I/O buffering. For information about how to tune these parameters, see Article ID 1072545.1: RMAN Performance Tuning Using Buffer Memory Parameters at [My Oracle Support](http://support.oracle.com).

For Oracle SPARC SuperCluster, tuning the following four parameters should be considered:

- `_backup_disk_bufcnt` - Number of buffers used to process backup sets
- `_backup_disk_bufsz` - Size of the buffers used to process backup sets
- `_backup_file_bufcnt` - Number of buffers used to process image copies
- `_backup_file_bufsz` - Size of the buffers used to process image copies

For backup and restore operations on backup sets and image copies, set the number of buffers to 64 and the buffer size to 1 MB:

```sql
SQL> alter system set "_backup_disk_bufcnt"=64;
SQL> alter system set "_backup_disk_bufsz"=1048576;
SQL> alter system set "_backup_file_bufcnt"=64;
SQL> alter system set "_backup_file_bufsz"=1048576;
```

These commands may be configured persistently by adding them to the SPFILE, or they may be set dynamically in the Oracle RMAN run block used to execute the backup or restore operations.

The following code fragments show how to dynamically tune the buffer sizes and counts for backup and restore operations.

- **Backup set backup:**

  ```sql
  run
  (alter system set "_backup_disk_bufcnt"=64;alter system set "_backup_disk_bufsz"=1048576;alter system set "_backup_file_bufcnt"=64;alter system set "_backup_file_bufsz"=1048576;allocate channel...); backup as backupset database;
  ```
Configuring Oracle Direct NFS (dNFS)

- **Backup set restore:**
  
  ```
  run
  {<br/>
  sql 'alter system set "_backup_disk_bufcnt"=64';<br/>  sql 'alter system set "_backup_disk_bufsz"=1048576';<br/>
  allocate channel...
  ...
  restore database;
  }
  ```

- **Image copy backup:**
  
  ```
  run
  {<br/>
  sql 'alter system set "_backup_file_bufcnt"=64';<br/>  sql 'alter system set "_backup_file_bufsz"=1048576';<br/>
  allocate channel...
  ...
  backup as copy database;
  }
  ```

- **Image copy restore:**
  
  ```
  run
  {<br/>
  sql 'alter system set "_backup_file_bufcnt"=64';<br/>  sql 'alter system set "_backup_file_bufsz"=1048576';<br/>
  allocate channel...
  ...
  restore database;
  }
  ```

Performing an incrementally applied backup requires reading an incremental backup set and writing to an image copy. To tune buffers for incrementally applied backups, run the following:

```
run
{<br/>
 sql 'alter system set "_backup_disk_bufcnt"=64';<br/>  sql 'alter system set "_backup_disk_bufsz"=1048576';<br/>  sql 'alter system set "_backup_file_bufcnt"=64';<br/>  sql 'alter system set "_backup_file_bufsz"=1048576';<br/>
 allocate channel...
 ...
 recover copy of database;
}
```
Creating Dedicated Services for Oracle RMAN Operations

Two services dedicated to Oracle RMAN processing can be configured to optimize management of load balancing, high availability, and upgrades. These services can be evenly load balanced over all the nodes of an Oracle SPARC SuperCluster system. Availability and performance can be optimized by configuring the services to run on a preferred instance while preparing them to fail over to any instance in the cluster. If these services are configured, upgrading a one-quarter or one-half rack Oracle SPARC SuperCluster system does not require changing the connect string of the Oracle RMAN run block.

The `srvctl` utility is used to install services for Oracle RMAN processing. The following code fragment shows how to create two services evenly distributed over a four-node cluster that are set up to fail over to any other node in the cluster. In this example, the services are installed for a database named `dbname` and are named `dbname_bkup`.

```
srvctl add service -d dbname -r dbname1 -a dbname2 -s dbname_bkup1
srvctl start service -d dbname -s dbname_bkup1
srvctl add service -d dbname -r dbname2 -a dbname1 -s dbname_bkup2
srvctl start service -d dbname -s dbname_bkup2
```

Configuring Oracle RMAN

Configuring Oracle RMAN channel and parallelism includes specifying the file system targets for the Oracle RMAN backup channels and the total number of channels used for backup and restore operations. Performance benefits can be realized by configuring 16 Oracle RMAN channels spanning the available ZFS Storage Appliance shares. Configure Oracle RMAN channels such that they are evenly distributed over the Oracle Database instances and nodes in the RAC cluster and evenly distributed over the shares exported from the ZFS Storage Appliance.

The following code fragments show sample Oracle RMAN run blocks for performing backup and restore operations for backup sets and image copies as well as applying incremental merges to image copies. The sample code is based on the following database configuration:

- Database name: `dbname`
- SYSDBA login: `sys/welcome`
- Scan address: `ad01-scan`
- Service names for the backup: `dbname_bkup`

The ZFS Storage Appliance can be configured in a one-pool configuration in which the appliance exports eight shares used as eight mount points.

The Oracle RMAN run blocks for backup and restore using backup sets and image copies are shown in the examples in the sections below. In these examples, the mount points for the four-share configuration are accessed as `/zfssa/dbname/backup1` through `/zfssa/dbname/backup4`. 
Also, the examples are for a configuration in which the ZFS Storage Appliance exports four shares used as four mount points for 16 Oracle RMAN channels.

Backup set level 0 backup:

```sql
run
{
  sql 'alter system set "_backup_disk_bufcnt"=64 scope=memory';
  sql 'alter system set "_backup_disk_bufsz"=1048576 scope=memory';
  allocate channel ch01 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup1' format '/zfssa/dbname/backup1/%U';
  allocate channel ch02 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup2' format '/zfssa/dbname/backup2/%U';
  allocate channel ch03 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup3' format '/zfssa/dbname/backup3/%U';
  allocate channel ch04 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup4' format '/zfssa/dbname/backup4/%U';
  allocate channel ch05 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup1' format '/zfssa/dbname/backup1/%U';
  allocate channel ch06 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup2' format '/zfssa/dbname/backup2/%U';
  allocate channel ch07 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup3' format '/zfssa/dbname/backup3/%U';
  allocate channel ch08 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup4' format '/zfssa/dbname/backup4/%U';
  allocate channel ch09 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup1' format '/zfssa/dbname/backup1/%U';
  allocate channel ch10 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup2' format '/zfssa/dbname/backup2/%U';
  allocate channel ch11 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup3' format '/zfssa/dbname/backup3/%U';
  allocate channel ch12 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup4' format '/zfssa/dbname/backup4/%U';
  allocate channel ch13 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup1' format '/zfssa/dbname/backup1/%U';
  allocate channel ch14 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup2' format '/zfssa/dbname/backup2/%U';
  allocate channel ch15 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup3' format '/zfssa/dbname/backup3/%U';
  allocate channel ch16 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup4' format '/zfssa/dbname/backup4/%U';
  configure snapshot controlfile name to '/zfssa/dbname/backup1.snapcf_dbname.f';
  backup as backupset incremental level 0 section size 32g database tag 'FULLBACKUPSET_L0' plus archivelog tag 'FULLBACKUPSET_L0';
}
```

Backup set level 1 backup:

```sql
run
{
  sql 'alter system set "_backup_disk_bufcnt"=64 scope=memory';
  sql 'alter system set "_backup_disk_bufsz"=1048576 scope=memory';
  allocate channel ch01 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup1' format '/zfssa/dbname/backup1/%U';
  allocate channel ch02 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup2' format '/zfssa/dbname/backup2/%U';
  allocate channel ch03 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup3' format '/zfssa/dbname/backup3/%U';
  allocate channel ch04 device type disk connect 'sys/welcome@ad01-<br/>
  scan/dbname_bkup4' format '/zfssa/dbname/backup4/%U';
  allocate channel ch05 device type disk connect 'sys/welcome@ad01-<br/>
sql 'alter system set "_backup_disk_bufsz"=1048576 scope=memory';
allocate channel ch01 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup1/%U';
allocate channel ch02 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup2/%U';
allocate channel ch03 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup3/%U';
allocate channel ch04 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup4/%U';
allocate channel ch05 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup1/%U';
allocate channel ch06 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup2/%U';
allocate channel ch07 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup3/%U';
allocate channel ch08 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup4/%U';
allocate channel ch09 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup2/%U';
allocate channel ch10 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup1/%U';
allocate channel ch11 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup4/%U';
allocate channel ch12 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup3/%U';
allocate channel ch13 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup4/%U';
allocate channel ch14 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup3/%U';
allocate channel ch15 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup4/%U';
allocate channel ch16 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup3/%U';
configure snapshot controlfile name to '/zfssa/dbname/backup1/snapcf_dbname.f';
backup as backupset incremental level 1 database tag 'FULLBACKUPSET_L1';
}

Image copy backup:

run
{<br/>
sql 'alter system set "_backup_file_bufcnt"=64 scope=memory';
sql 'alter system set "_backup_file_bufsz"=1048576 scope=memory';
allocate channel ch01 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup1/%U';
allocate channel ch02 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup2/%U';
allocate channel ch03 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup3/%U';
allocate channel ch04 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup4/%U';
allocate channel ch05 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup2/%U';
allocate channel ch06 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup1/%U';
allocate channel ch07 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup3/%U';
allocate channel ch08 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup4/%U';
allocate channel ch09 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup2/%U';
allocate channel ch10 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup1/%U';
allocate channel ch11 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup4/%U';
allocate channel ch12 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup3/%U';
allocate channel ch13 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup4/%U';
allocate channel ch14 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup3/%U';
allocate channel ch15 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1' format '/zfssa/dbname/backup4/%U';
allocate channel ch16 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2' format '/zfssa/dbname/backup3/%U';
configure snapshot controlfile name to '/zfssa/dbname/bkup1/snapcf_dbname.f';
backup as backupset incremental level 1 database tag 'FULLBACKUPSET_L1';
}
scan/dbname_bkup1' format '/zfssa/dbname/backup1/%U';<br/>
allocate channel ch06 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup2' format '/zfssa/dbname/backup2/%U';<br/>
allocate channel ch07 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup1' format '/zfssa/dbname/backup3/%U';<br/>
allocate channel ch08 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup2' format '/zfssa/dbname/backup4/%U';<br/>
allocate channel ch09 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup1' format '/zfssa/dbname/backup2/%U';<br/>
allocate channel ch10 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup2' format '/zfssa/dbname/backup1/%U';<br/>
allocate channel ch11 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup1' format '/zfssa/dbname/backup4/%U';<br/>
allocate channel ch12 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup2' format '/zfssa/dbname/backup3/%U';<br/>
allocate channel ch13 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup1' format '/zfssa/dbname/backup2/%U';<br/>
allocate channel ch14 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup2' format '/zfssa/dbname/backup1/%U';<br/>
allocate channel ch15 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup1' format '/zfssa/dbname/backup4/%U';<br/>
allocate channel ch16 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup2' format '/zfssa/dbname/backup3/%U';<br/>
configure snapshot controlfile name to<br/>
'/zfssa/dbname/backup1/snapcf_dbname.f';<br/>
backup incremental level 1 for recover of copy with tag 'IMAGECOPY'<br/>
database;
}

Incremental merge to image copy:

run
{
sql 'alter system set "_backup_disk_bufcnt"=64 scope=memory';<br/>
sql 'alter system set "_backup_disk_bufsz"=1048576 scope=memory';<br/>
sql 'alter system set "_backup_file_bufcnt"=64 scope=memory';<br/>
sql 'alter system set "_backup_file_bufsz"=1048576 scope=memory';<br/>
allocate channel ch01 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup1';<br/>
allocate channel ch02 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup2';<br/>
allocate channel ch03 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup1';<br/>
allocate channel ch04 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup2';<br/>
allocate channel ch05 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup1';<br/>
allocate channel ch06 device type disk connect 'sys/welcome@ad01-<br/>
scan/dbname_bkup2';<br/>
Configuring Oracle Direct NFS (dNFS)

Integrating Applications with the Oracle ZFS Storage Appliance

Scan/dbname_bkup1';
allocate channel ch10 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2';
allocate channel ch11 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1';
allocate channel ch12 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2';
allocate channel ch13 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1';
allocate channel ch14 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2';
allocate channel ch15 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1';
allocate channel ch16 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2';

Configure snapshot controlfile name to:
'/zfssa/dbname/backup1/snapcf_dbname.f';
recover copy of database with tag 'IMAGECOPY';

Restore validate:

run
{<br/>
sql 'alter system set "_backup_disk_bufcnt"=64 scope=memory';
sql 'alter system set "_backup_disk_bufsz"=1048576 scope=memory';
sql 'alter system set "_backup_file_bufcnt"=64 scope=memory';
sql 'alter system set "_backup_file_bufsz"=1048576 scope=memory';
allocate channel ch01 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1';
allocate channel ch02 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2';
allocate channel ch03 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1';
allocate channel ch04 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2';
allocate channel ch05 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1';
allocate channel ch06 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2';
allocate channel ch07 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1';
allocate channel ch08 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2';
allocate channel ch09 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1';
allocate channel ch10 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2';
allocate channel ch11 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1';
allocate channel ch12 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup2';
allocate channel ch13 device type disk connect 'sys/welcome@ad01-
scan/dbname_bkup1';
}
Setting Up the Directory Structure to Mount the Shares on the Host

allocate channel ch14 device type disk connect \\
scan/dbname_bkup2';<br/>
allocate channel ch15 device type disk connect \\
scan/dbname_bkup1';<br/>
allocate channel ch16 device type disk connect \\
scan/dbname_bkup2';<br/>
configure snapshot controlfile name to \\
'/zfssa/dbname/backup1/snapcf_dbname.f';<br/>
restore validate database;

Configuring Oracle SPARC SuperCluster for Appliance Backup

Use the following procedures to configure the Oracle SPARC SuperCluster for Appliance backup.

■ “Configuring the Appliance for SuperCluster Backup” on page 424
■ “Setting Up the Directory Structure to Mount the Shares on the Host” on page 442
■ “Updating the /etc/vfstab File” on page 442
■ “Enabling the NFS Client Service” on page 443
■ “Updating oranfstab to Access ZFS Storage Appliance Exports” on page 443
■ “Mounting the Shares on the Host” on page 444
■ “Setting the Ownership of the Mounted Shares” on page 444

▼ Setting Up the Directory Structure to Mount the Shares on the Host

Set up mount points for the shares on the host as shown:

```bash
mkdir -p /zfssa/dbname/backup1
mkdir -p /zfssa/dbname/backup2
mkdir -p /zfssa/dbname/backup3
mkdir -p /zfssa/dbname/backup4
```

▼ Updating the /etc/vfstab File

The UNIX new-line escape character (\) indicates a single line of code has been wrapped to a second line in the listing below. When entering a wrapped line into fstab, remove the \ character and combine the two line segments, separated by a space, into a single line.

1. To update the /etc/vfstab file, use one of the following options.
For a one-pool configuration:

```
192.168.36.200:/export/dbname/backup1 - /zfssa/dbname/backup1 
  nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= tcp,vers=3,forcedirectio
192.168.36.200:/export/dbname/backup2 - /zfssa/dbname/backup2 
  nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= tcp,vers=3,forcedirectio
192.168.36.200:/export/dbname/backup3 - /zfssa/dbname/backup3 
  nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= tcp,vers=3,forcedirectio
192.168.36.200:/export/dbname/backup4 - /zfssa/dbname/backup4 
  nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= tcp,vers=3,forcedirectio
```

2. For a two-pool configuration:

```
192.168.36.200:/export/dbname/backup1 - /zfssa/dbname/backup1 
  nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= tcp,vers=3,forcedirectio
192.168.36.201:/export/dbname/backup2 - /zfssa/dbname/backup2 
  nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= tcp,vers=3,forcedirectio
192.168.36.200:/export/dbname/backup3 - /zfssa/dbname/backup3 
  nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= tcp,vers=3,forcedirectio
192.168.36.201:/export/dbname/backup4 - /zfssa/dbname/backup4 
  nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= tcp,vers=3,forcedirectio
```

► Enabling the NFS Client Service

- Enable the NFS Client Service on the Solaris 11 host with the following command:

```
svcadm enable -r nfs/client
```

► Updating oranfstab to Access ZFS Storage Appliance Exports

To update the oranfstab file to access ZFS Storage Appliance exports, use the appropriate following option.

1. For a one-pool configuration:

```
server: 192.168.36.200
path: 192.168.36.200
path: 192.168.36.201
```
Mounting the Shares on the Host

For a two-pool configuration:

server: 192.168.36.200
path: 192.168.36.200
path: 192.168.36.202
export: /export/dbname/backup1 mount: /zfssa/dbname-2pool/backup1
export: /export/dbname/backup3 mount: /zfssa/dbname-2pool/backup3
server: 192.168.36.201
path: 192.168.36.201
path: 192.168.36.203
export: /export/dbname/backup2 mount: /zfssa/dbname-2pool/backup2
export: /export/dbname/backup4 mount: /zfssa/dbname-2pool/backup4

Mounting the Shares on the Host

Using the standard Solaris `mount` command, manually mount the shares:

```
# mount /zfssa/dbname/backup1
# mount /zfssa/dbname/backup2
# mount /zfssa/dbname/backup3
# mount /zfssa/dbname/backup4
```

Setting the Ownership of the Mounted Shares

Change the permission settings of the mounted shares to match the permission settings of `ORACLE_HOME`. In this example, the user and group ownerships are set to `oracle:dba`.

1. Enter:
   ```bash
   # chown oracle:dba /zfssa/dbname/*
   ```

2. Restart the Oracle Database instance to pick up the changes that were made to the `oranfstab` file using one of the following options:
   - Restart one instance at a time (rolling upgrade), for example:
     ```bash
     $ srvctl stop instance -d dbname -i dbname1
     $ srvctl start instance -d dbname -i dbname1
     $ srvctl stop instance -d dbname -i dbname2
     $ srvctl start instance -d dbname -i dbname2
     $ srvctl stop instance -d dbname -i dbname3
     ```
Oracle Intelligent Storage Protocol

Oracle Intelligent Storage Protocol (OISP) enables the Oracle direct NFS (dNFS) client to encode and pass attributes associated with I/O requests to the Oracle ZFS Storage Appliance. These attributes contain such information as the type of database file that the I/O request is targeting, the record size of the file, and the identity of the database issuing the I/O request.

The Oracle ZFS Storage Appliance decodes these attributes, using them to simplify database configuration, increase database performance, and provide observability into the source of I/O workloads being generated by database clients.

Database Record Size

The Oracle dNFS client can pass the optimal record size based on the type of file for each I/O request. If a record size is passed, it overrides the "Database record size" property setting on the share or project. The record size can only be set for newly created files. If a file already exists, the record size is not changed.

Synchronous Write Bias Hint

The Oracle dNFS client can pass a write bias "hint" associated with write I/O requests that prompts the Oracle ZFS Storage Appliance to treat I/O requests as either latency sensitive or
throughput oriented. If the hint is passed, it overrides the "Synchronous write bias" property setting on the share or project.

Analytics Breakdown by Database Name

The Oracle 12c dNFS client can pass the identification of the database (SID) or container database and pluggable database (SID:SID) responsible for issuing I/O requests. Oracle ZFS Storage Appliance analytics can display I/O statistics broken down by the SID name(s) of the database by selecting breakdown or drill by "Application ID."

OISP-Capable Protocols and Clients

Protocols: NFSv4

Clients: Oracle Database NFS (dNFS) client

Appliance Network File System Plug-in for Oracle Solaris Cluster

Oracle Solaris Cluster (OSC) is a high-availability cluster software product for the Solaris Operating System.

The appliance Network File System Plug In for Oracle Solaris Cluster enables OSC with the appliance using NFS protocol. The plug-in and readme file are available as part of the Oracle ZFS Storage Appliance Network File System Plug-in for Oracle Solaris Cluster on the Oracle Technology Network.

Appliance Plug-in for Oracle Solaris Cluster Geographic Edition

Oracle Solaris Cluster Geographic Edition software is a layered extension of the Oracle Solaris Cluster software. The Geographic Edition software protects applications from unexpected disruptions by using multiple clusters that are separated by long distances, and by using a redundant infrastructure that replicates data between these cluster sites. This plug-in coordinates data replication between remote Oracle Solaris Cluster sites using the appliance remote replication service.
The plug-in package is available through the Oracle Technology Network Sun NAS Storage information page.

**Appliance Plug-in for Oracle Enterprise Manager System Monitoring**

The Oracle ZFS Storage Plug-in for Oracle Enterprise Manager System Monitoring provides first-class monitoring to the grid controller environment for the Oracle ZFS Storage Appliance. The plug-in has the ability to:

- Monitor Oracle ZFS Storage appliances
- Gather storage system information, configuration information and performance information of accessible storage components
- Raise alerts and violations based on thresholds and monitoring information collected by the tool
- Provide out-of-the-box reports that complement analytics
- Support monitoring by remote agents

Once an appliance is configured to be monitored by the grid controller, analytics worksheets and datasets are created to bridge the grid controller administrator's view to the deeper level of detail provided by the real-time analytics available within the appliance.

The management plug-in is available at the following link: [http://www.oracle.com/technetwork/documentation/index.html](http://www.oracle.com/technetwork/documentation/index.html).

It is packaged with an installation guide that should be read by both administrators of the grid controller and storage administrators of appliances being monitored.

Included with each appliance are two workflows that are used respectively to prepare a system for monitoring, or to remove the artifacts created for the monitoring environment:

- Configure for Oracle Enterprise Manager Monitoring
- Unconfigure Oracle Enterprise Manager Monitoring

These workflows are accessible from the Maintenance > Workflows page in the browser user interface.

**Configuring for Oracle Enterprise Manager Monitoring**

This workflow is used to prepare an environment for monitoring, or to reset any of the artifacts that were created by the workflow back to their original state in the event the artifacts were...
changed during operation by the storage administrator. Executing this workflow makes the following changes to the system:

- An oracle_agent Role Properties will be created with limited access to the system, to allow the Oracle Enterprise Manager Grid Controller agent to obtain information required for monitoring, but not to make alterations to the system. An oracle_agent user will be created and assigned this role. Use of this role and user is critical to keeping clean audit records for when and how the agent accesses the appliance.
- Advanced Analytics will be enabled, makes an extended set of statistics available to all users of the Oracle ZFS Storage appliance.
- The Worksheet Oracle Enterprise Manager will be created, facilitating communication between the grid controller administrator and the storage administrator. All metrics monitored by grid controller are available from this worksheet.

Unconfiguring Oracle Enterprise Manager Monitoring

This workflow removes artifacts created by Configure for Oracle Enterprise Manager Monitoring. Specifically, it:

- Removes the oracle_agent role and user
- Removes the Oracle Enterprise Manager worksheet

This workflow will not disable Advanced Analytics or any of the datasets that were activated for collection purposes.

Appliance Plug-in for Oracle Virtual Machine Storage Connect

One of many new features introduced within Oracle VM 3.0 is the Storage Connect framework. This framework enables Oracle VM 3.0 Manager to directly access storage servers and provision resources. With this framework, you can register storage servers, discover existing storage resources, create and present physical disks to server pools, and share storage repositories and Virtual Machines.

The Oracle Virtual Machine Storage Connect Plug-in for the Oracle ZFS Storage Appliance is a component of the Oracle VM software suite that enables Oracle VM to provision and manage the Oracle ZFS Storage Appliance for virtualization. The plug-in is installed on the Oracle VM Server(s) and communicates with the storage server(s) through workflows installed on the appliance.

The plug-in and readme file are available on the Oracle Technology Network.
Appliance Plug-in Provider for Volume Shadow Copy Service Software

Volume Shadow Copy Services (VSS) for Microsoft operating systems provides a framework to allow volume backups to be performed while applications on a system continue to write to the volumes. VSS provides a consistent interface that allows coordination between user applications that update data on disk (VSS writers) and those that back up applications (VSS requesters). Specifically, VSS provides:

- A backup infrastructure that coordinates applications with file system activities
- A location to create point in time, coalesced copies known as shadow copies

The Oracle ZFS Storage Appliance Provider for Volume Shadow Copy Service Software is a VSS hardware provider that allows the Oracle ZFS Storage Appliance to take consistent snapshots for Windows hosts which are using block targets. VSS coordinates snapshots to ensure block data is consistent. The provider communicates with a set of workflows on the appliance to coordinate taking of snapshots as seen from the application. It works over both iSCSI and Fibre Channel.

The Oracle ZFS Storage Appliance Provider for Volume Shadow Copy Service Software is installed on hosts that require this functionality and coordination between applications. Complete documentation for this application integration is packaged with the downloaded components in the form of a readme file. The provider software and readme file are available as part of the Oracle ZFS Storage Appliance Software Providers and Plug-ins patch on the Oracle Technology Network. More information on VSS is available on the Microsoft web site: http://msdn.microsoft.com/en-us/library/aa384649

FC Support with Symantec Dynamic Multi-Pathing and Storage Foundation

- **SF** - Symantec Storage Foundation 5.1
- **SF HA** - Storage Foundation High Availability 5.1
- **SFCFS/SF Oracle RAC** - Storage Foundation Cluster File System/Storage Foundation for Oracle RAC 5.1
- **SFCFS/SFCFS Oracle RAC** - Storage Foundation Cluster File System/Storage Foundation Cluster File System for Oracle RAC 5.1
- Solaris 10 SPARC
- Solaris 10 x86
- Linux RedHat5
- Oracle Enterprise Linux (OEL)

Note the following restrictions:

- Symantec's "required" appliance ASLs be installed which can be downloaded from: https://vos.symantec.com/asl
- Symantec also required SF 5.1 VM patch level of 5.1RP2 or greater which can be downloaded from: https://vos.symantec.com/patch/matrix
- Symantec also requires the following DMP parameter setting (only for 'clustered' appliances) of:
  - :dmp_health_time=0
  - :dmp_path_age=0
  - :dmp_lun_retry_timeout=200

Refer to Symantec's HW tech note which references the 'clustered' appliance settings: Symantec Technical Support (http://www.symantec.com/business/support/index?page=content&id=TECH74012).

Symantec's Storage Foundation 5.1SP2 for Windows supports FC connections to the Oracle ZFS Storage Appliance for the following Windows versions:

- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2

Refer to the SF 5.1SP2 HCL at Hardware Compatibility List (HCL) for Symantec Storage Foundation (http://www.symantec.com/business/support/index?page=content&id=TECH138719)

Appliance Replication Adapter for VMware Site Recovery Manager

The Oracle ZFS Storage Appliance Storage Replication Adapter (SRA) for VMware vCenter Site Recovery Manager (SRM) integrates the Oracle ZFS Storage Appliance into VMware deployments that span multiple sites and require fast recovery in the event of a protected site service disruption. The SRA plugs into existing VMware vCenter SRM environments and allows the Oracle ZFS Storage Appliance to be managed through VMware vCenter SRM discovery, test, and failover sequences as the recovery plan is tested and run. Usage of the SRA occurs entirely within the VMware vCenter SRM application.

The VMware administrator will need to work closely with the Oracle ZFS Storage Appliance administrator responsible for the appliance that hosts the VMware data stores. For further
Appliance Replication Adapter for VMware Site Recovery Manager

information, see the Oracle ZFS Storage Appliance SRA for VMware SRM Administration Guide that is packaged in the SRA.

Note - The SRA can be downloaded from the Oracle Technology Network. A valid Oracle support contract for the Oracle ZFS Storage Appliance is required to obtain the SRA.