Sun ZFS Storage 7000 System Administration Guide

Part No: 820–4167–13 Rev. A
November 2010

101109@24999

Contents

Preface 19

1 **Introduction** 21
Overview 21
Introduction 21
User Interface 24
Browser User Interface (BUI) 25
Command Line Interface (CLI) 25
Browsers 25
Supported Browsers 25
Main Window 26
Overview 26
Masthead 26
Title Bar 27
Side Panels and Menu Titles 28
Non-Standard BUI Control Primer 28
Icons 31
General Usage 31
CLI 35
CLI Introduction 35
Contexts 36
CLI Contexts 36
Returning to a Previous Context 37
Navigating to a Parent Context 38
Contexts and Tab-Completion 38
Executing Context-Specific Commands 39
Uncommitted Contexts 39
Properties 40

CLI Properties 40
Getting Properties 40
Setting Properties 41
Immutable Properties 42
Scripting 43
Batching Commands 43
Scripting 43
Automating Access 49

2 Status 51
Status 51
Introduction 51
Dashboard 52
Links 52
CLI 57
Tasks 58
Settings 58
Introduction 58
BUI 58
CLI 60
Tasks 60
NDMP 61
BUI 61
CLI 62

3 Configuration 63
Configuration 63
Introduction 63
Initial 64
Initial Configuration 64
Prerequisites 64
Summary 64
BUI 65
CLI 66
Network 70

Network Configuration 71
BUI 78
CLI 80
Tasks 82
Storage 87
Introduction 87
Tasks 91
SAN 92
SAN 92
BUI 93
CLI 95
Terms 95
SAN Terminology 95
FC 97
Fibre Channel 97
BUI 100
CLI 102
FCMPxIO 105
Configuring FC Client Multipathing 105
iSCSI 109
Introduction 109
BUI 112
CLI 113
SRP 114
Introduction 114
BUI 121
CLI 124
Users 125
Introduction 125
Roles 125
Authorizations 126
Properties 127
BUI 128
CLI 128
Tasks 130
Preferences 133

Introduction 133
BUI 133
CLI 134
SSH Public Keys 134
Alerts 135
Introduction 135
Actions 136
Threshold Alerts 137
BUI 138
CLI 138
Tasks 138
Workflows 139
Introduction 139
BUI 151
CLI 151
Cluster 153
Clustering 153
Features and Benefits 154
Drawbacks 155
Terminology 156
Subsystem Design 156
Configuration Changes in a Clustered Environment 162
Clustering Considerations for Storage 163
Clustering Considerations for Networking 164
Clustering Considerations for Infiniband 165
Preventing "Split-Brain" Conditions 167
Estimating and Reducing Takeover Impact 170
Setup Procedure 171
Node Cabling 173
JBOD Cabling 174
BUI 174
Unconfiguring Clustering 175

4 Services 177
Services 177

Introduction 177
BUI 179
CLI 181
NFS 184
Introduction 184
Properties 184
Kerberos realms 185
Logs 186
Analytics 186
CLI 186
Tasks 187
iSCSI 188
Introduction 188
Properties 188
Authentication 188
Authorization 188
Targets and Initiators 189
CLI 189
Tips 189
SMB 189
Introduction 189
Properties 190
Share Properties 191
NFS/SMB Interoperability 191
DFS Namespaces 191
Autohome Rules 192
Local Groups 192
MMC Integration 193
CLI 198
Tasks 199
FTP 202
Introduction 202
Properties 202
Logs 203
Tasks 204
HTTP 204

Introduction 204
Properties 204
Authentication and Access Control 205
Logs 205
Tasks 206
NDMP 206
Introduction 206
Properties 211
Logs 213
SFTP 213
Introduction 213
Properties 213
Logs 214
Tasks 214
Virus Scan 214
Introduction 214
Properties 215
Logs 216
Tasks 216
NIS 217
Introduction 217
Properties 217
Logs 217
Tasks 218
LDAP 218
Introduction 218
Properties 219
Logs 220
Tasks 220
Active Directory 221
Introduction 221
Properties 221
Domains and Workgroups 222
LDAP Signing 222
Windows Server 2008 Support 223
BUI 224

CLI 224
Tasks 225
Identity Mapping 226
Concepts 226
IDMU 227
Directory-based Mapping 227
Name-based Mapping 228
Ephemeral Mapping 230
Best Practices 230
Testing Mappings 231
Examples 231
Tasks 232
DNS 233
Introduction 233
Properties 233
CLI 233
Logs 234
Active Directory and DNS 234
Non-DNS Resolution 234
DNS-Less Operation 235
IPMP 235
Introduction 235
Properties 235
Logs 236
Tasks 236
NTP 236
Introduction 236
Properties 236
BUI Clock 238
Tips 238
Tasks 239
Remote Replication 239
Introduction 239
Dynamic Routing 240
RIP and RIPng Dynamic Routing Protocols 240
Logs 240

Phone Home 240
Introduction 240
Properties 241
Service state 242
Logs 242
SNMP 242
Introduction 242
Properties 243
MIBs 243
Sun FM MIB 244
Sun AK MIB 244
Tasks 245
SMTP 245
Introduction 245
Properties 246
Logs 246
Service Tags 246
Introduction 246
Properties 247
System Identity 247
Introduction 247
Properties 247
Logs 248
SSH 248
Introduction 248
Properties 248
Logs 248
Tasks 249
Shadow Migration 249
Introduction 249
Properties 249
Managing Shadow Migration 250
Syslog 250
Introduction 250
Properties 251
Classic Syslog: RFC 3164 251

Updated Syslog: RFC 5424 251
Message Format 251
Receiver Configuration Examples 253

5 Shares 255
Shares 255
Introduction 255
Concepts 256
Storage Pools 256
Projects 257
Shares 257
Properties 257
Snapshots 258
Clones 259
Shadow Migration 259
Shadow Data Migration 259
Shadow migration behavior 262
Shadow Migration Management 263
Migration of local filesystems 267
Tasks 267
Space Management 268
Introduction 268
Terms 268
Understanding snapshots 270
Filesystem and project settings 271
User and group settings 272
Filesystem Namespace 275
Filesystem namespace 275
Shares 277
BUI 277
CLI 282
General 287
General Share Properties 287
Space Usage 287
Properties 288

Custom Properties 293
Protocols 293
Shares Protocols 293
NFS 294
SMB 298
SCSI 298
HTTP 299
FTP 299
SFTP 300
Access 300
Access Control 300
Root Directory Access 300
ACL Behavior 301
Root Directory ACL 303
Snapshots 305
Introduction 305
Snapshot Properties 305
BUI 306
CLI 309
Projects 311
BUI 311
CLI 313
General 317
General Project Properties 317
Space Usage 317
Inherited Properties 318
Custom Properties 318
Filesystem Creation Defaults 318
LUN Creation Defaults 318
Protocols 319
Project Protocols 319
NFS 319
SMB 319
iSCSI 319
HTTP 320
FTP 320

Access 320
Access Control 320
Inherited ACL Behavior 320
Snapshots 320
Introduction 320
Snapshot Properites 320
BUI 321
CLI 321
Replication 321
Remote Replication Introduction 321
Concepts 323
Configuring Replication 327
Managing Replication Packages 330
Remote Replication Details 338
Schema 345
Customized Share Properties 345
BUI 345
CLI 346
Tasks 347

6 Analytics 349
Analytics 349
Introduction 350
Concepts 350
Analytics 350
Drilldown Analysis 350
Statistics 351
Datasets 352
Actions 352
Worksheets 352
Statistics 353
Introduction 353
Descriptions 353
Default Statistics 354
Tasks 356

CPU Percent utilization 357
CPU: Percent Utilization 357
Cache ARC accesses 358
Cache: ARC accesses 358
Cache L2ARC IO bytes 361
Cache: L2ARC I/O bytes 361
Cache L2ARC accesses 361
Cache: L2ARC accesses 361
Data Movement NDMP bytes transferred tofrom disk 362
Data Movement: NDMP bytes transferred to/from disk 362
Data Movement NDMP bytes transferred tofrom tape 363
Data Movement: NDMP bytes transferred to/from tape 363
Data Movement Shadow migration bytes 363
Data Movement: Shadow migration bytes 363
Data Movement Shadow migration ops 364
Data Movement: Shadow migration ops 364
Data Movement Shadow migration requests 365
Data Movement: Shadow migration requests 365
Disk Disks 366
Disk: Disks 366
Disk IO bytes 367
Disk: I/O bytes 367
Disk IO operations 368
Disk: I/O operations 368
Network Device bytes 370
Network: Device bytes 370
Network Interface bytes 370
Network: Interface bytes 370
Protocol SMB operations 371
Protocol: SMB operations 371
Protocol Fibre Channel bytes 373
Protocol: Fibre Channel bytes 373
Protocol Fibre Channel operations 374
Protocol: Fibre Channel operations 374
Protocol FTP bytes 375
Protocol: FTP bytes 375

Protocol HTTPWebDAV requests 376
Protocol: HTTP/WebDAV requests 376
Protocol iSCSI bytes 378
Protocol: iSCSI bytes 378
Protocol iSCSI operations 378
Protocol: iSCSI operations 378
Protocol NFSv2 operations 380
Protocol: NFSv2 operations 380
Protocol NFSv3 operations 382
Protocol: NFSv3 operations 382
Protocol NFSv4 operations 383
Protocol: NFSv4 operations 383
Protocol SFTP bytes 385
Protocol: SFTP bytes 385
Protocol SRP bytes 386
Protocol: SRP bytes 386
Protocol SRP operations 387
Protocol: SRP operations 387
CPU CPUs 388
CPU: CPUs 388
CPU Kernel spins 389
CPU: Kernel spins 389
Cache ARC adaptive parameter 390
Cache: ARC adaptive parameter 390
Cache ARC evicted bytes 390
Cache: ARC evicted bytes 390
Cache ARC size 391
Cache: ARC size 391
Cache ARC target size 392
Cache: ARC target size 392
Cache DNLC accesses 393
Cache: DNLC accesses 393
Cache DNLC entries 393
Cache: DNLC entries 393
Cache L2ARC errors 394
Cache: L2ARC errors 394

Cache L2ARC size 395
Cache: L2ARC size 395
Data Movement NDMP file system operations 395
Data Movement: NDMP file system operations 395
Data Movement NDMP jobs 396
Data Movement: NDMP jobs 396
Disk Percent utilization 396
Disk: Percent utilization 396
Disk ZFS DMU operations 397
Disk: ZFS DMU operations 397
Disk ZFS logical IO bytes 398
Disk: ZFS logical I/O bytes 398
Disk ZFS logical IO operations 398
Disk: ZFS logical I/O operations 398
Memory Dynamic memory usage 399
Memory: Dynamic memory usage 399
Memory Kernel memory 400
Memory: Kernel memory 400
Memory Kernel memory in use 400
Memory: Kernel memory in use 400
Memory Kernel memory lost to fragmentation 401
Memory: Kernel memory lost to fragmentation 401
Network IP bytes 401
Network: IP bytes 401
Network IP packets 402
Network: IP packets 402
Network TCP bytes 403
Network: TCP bytes 403
Network TCP packets 403
Network: TCP packets 403
System NSCD backend requests 404
System: NSCD backend requests 404
System NSCD operations 405
System: NSCD operations 405
Open Worksheets 405
Worksheets 406

Saving a Worksheet 409
Toolbar Reference 409
CLI 411
Tips 411
Tasks 411
Saved Worksheets 412
Introduction 412
Properties 413
BUI 413
CLI 413
Datasets 414
Introduction 414
BUI 415
CLI 415

7 Application Integration 419
Application Integration 419
Introduction 419
Microsoft 420
Sun Storage 7000 Provider for Microsoft VSS Software 420
Oracle 421
Sun Storage 7000 Management Plug-In for Oracle Enterprise Manager 10g Grid Controller 421

Glossary 425

Index 429

Preface

The *Sun ZFS Storage 7000 System Administration Guide* contains administration and configuration documentation for Oracle's Sun ZFS Storage 7000 series of NAS appliances.

This documentation is also available while using the appliance Browser User Interface, accessible via the Help button. The appliance documentation may be updated using the System Upgrade procedure documented in the System Service Manual.

Who Should Use This Book

These notes are for users and system administrators who install and use the Sun ZFS Storage 7000 Appliances.

Related Documentation

Refer to the following documentation for installation instructions, hardware overviews, service procedures and software update notes.

- Installation Guide, Analytics Guide and Service Manual (`http://wikis.sun.com/display/fishworks/documentation/`)
- Release Notes (`http://wikis.sun.com/display/fishworks/software+updates`)

Third-Party Web Site References

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

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

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.education.oracle.com`)

◆ ◆ ◆ CHAPTER 1

Introduction

Overview

Introduction

The Sun ZFS Storage 7000 family of products provide 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.

Platforms

- 7120
- 7320
- 7420/7720
- 7700

Legacy platforms are documented in the 7110, 7210, 7310, 7410, J4400/J4500 sections of the Sun Storage 7000 Unified Storage System Service Manual available at http://wikis.sun.com/display/fishworks. (`http://wikis.sun.com/display/fishworks.`)

Expansion Storage

- Sun Disk Shelf

Protocols

Sun ZFS Storage appliances include support for a variety of industry-standard client protocols, including:

- SMB
- NFS
- HTTP and HTTPS
- WebDAV
- iSCSI
- FC
- SRP
- iSER
- FTP
- SFTP

Key Features

Sun ZFS Storage systems also include new 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

Data Services

To manage the data that you export using these protocols, you can configure your Sun ZFS Storage system using the built-in collection of advanced data services, including:

- RAID-Z (RAID-5 and RAID-6), mirrored, and striped disk configurations
- Unlimited read-only and read-write snapshots, with snapshot schedules
- Data deduplication
- Built-in data compression
- Remote replication of data for disaster recovery
- Active-active clustering for high availability (7310, 7320, 7410, 7420, and 7720)
- Thin provisioning of iSCSI LUNs
- Virus scanning and quarantine
- NDMP backup and restore

Availability

To maximize the availability of your data in production, Sun ZFS Storage appliances include 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 (7310, 7320, 7410, 7420, and 7720)
- Link aggregations and IP multipathing for network failure protection
- I/O Multipathing between the controller and disk shelves
- Integrated software restart of all system software services
- Phone-Home of telemetry for all software and hardware issues
- Lights-out Management of each system for remote power control and console access

Browser User Interface (BUI)

The browser user interface

The 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 management software is designed to be fully featured and functional on the following supported web browsers: Firefox 3.x, Internet Explorer 7 and 8, Safari 3.1 or later, and WebKit 525.13 or later.

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.

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 following sections describe details of the CLI. 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
```

User Interface

The browser user interface

Browser User Interface (BUI)

The 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 following sections provide and overview of the BUI.

- Main Window - overview of BUI elements and design
- Icons - icon reference
- Browsers - supported browsers

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 following sections describe details of the CLI.

- CLI - usage
- Contexts - contexts
- Properties - properties
- Scripting - scripting

Browsers

Supported Browsers

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

Tier 1

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

- Firefox 3.x
- Internet Explorer 7
- Internet Explorer 8
- Safari 3.1 or later
- WebKit 525.13 or later

Tier 2

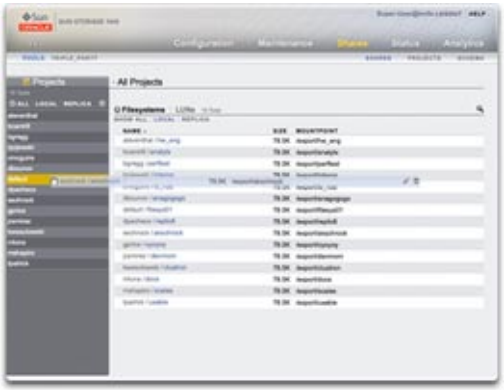
BUI elements may be cosmetically imperfect in tier 2 browsers, and some functionality may not be available, although all necessary features work correctly. A warning message appears during login if you are using one of the following tier 2 browser:

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

Unsupported Browsers

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

Main Window



Changing a filesystem's properties by moving it into another project using the Projects side panel.

Overview

The BUI provides an uncluttered environment for visualizing system behavior and identifying performance issues with the appliance.

Masthead

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

Navigation

Use 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.

Alerts

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 Maintenance: Logs screen.

Session Annotation

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. See Configuration: Users for details about session annotations.

Title Bar

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

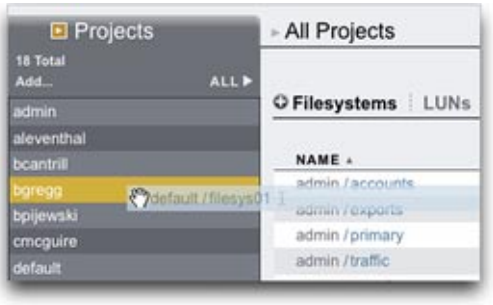
For example, the Identity mapping service title bar enables the following:

- 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

Side Panels and Menu Titles

To quickly navigate between Service and Project views, open and close the side panel by clicking the title or the reveal arrow.



Main Window Side Panels and Menu Titles

Add Projects

To add projects, click the Add... link in the sidebar.

Move Shares

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.

Object Name

To change a Share name, click the rename icon in the highlighted table row for the Share.

Non-Standard BUI Control Primer

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

Summary of BUI Controls	
Modify a property	Click the edit icon and complete the dialog
Add a list item or property entry	Click the add icon
Remove a list item or property entry	Click the remove icon
Save changes	Click the Apply button
Undo saved changes	Click the Revert button
Delete an item from a list	Click the trash icon (hover the mouse over the item row to see the icon)
Search for an item in a list	Click the search icon at the top right of the list
Sort by list headings	Click on the bold sub-headings to re-sort the list
Move or drag an item	Click the move icon
Rename an item	Click the rename icon
View details about your system	Oracle logo or click the model badge to go to the oracle.com web page for your mod
Automatically open side panel	Drag an item to the side panel
Send BUI feedback	Click the Let us know link at the bottom right of any screen to send us your suggestions about the interface or any other aspect of the appliance

Permissions

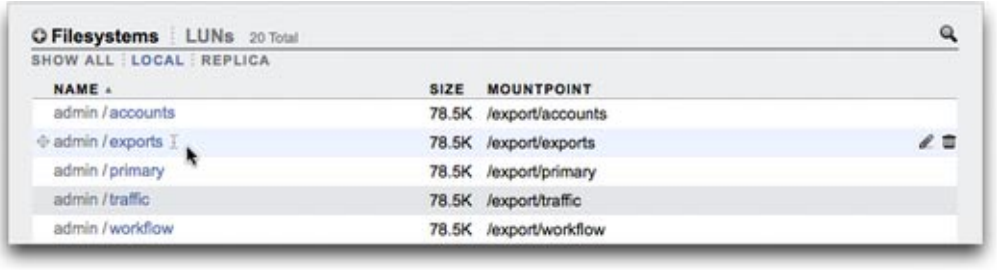
When setting permissions, the RWX boxes are clickable targets. Clicking on the access group label (User, Group, Other) toggles all permissions for that label on and off.

Editing Share Properties

To edit Share properties, deselect the Inherit from project checkbox.

Viewing List Item Controls

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



Modal Dialogs

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

General Usage

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.

Status

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

Icon	Description	Icon	Description
	on		warning
	off		disabled

Basic Usage

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

Icon*		Description	Icon*		Description
--		rename (edit text)	--		clone
--		move	--		rollback
--		edit	--		appliance power
--		destroy	--		appliance restart
		add	--		apply

Icon*		Description	Icon*	Description
		remove	--	revert
		cancel/close	--	info
--		error	--	sort list column (down)
--		alert	--	sort list column (up)
		on/off toggle		first page
		restart		previous page
--		locate		next page
		disable/offline		last page
		lock	--	search
--		wait spinner		menu
--		reverse direction		panel
--		sever		

** Disabled icons are shown at left.*

Networking

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

Icon	Description	Icon	Description
	active network device		active Infiniband port
	inactive network device		inactive Infiniband port
	network datalink		network datalink (IB partition)
	network datalink VLAN		
	network datalink aggregation		
	network datalink aggregation VLAN		

Dashboard Thresholds

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

Icon	Description	Icon	Description
	sunny		hurricane
	partly cloudy		hurricane class 2
	cloudy		hurricane class 3
	rainy		hurricane class 4
	stormy		hurricane class 5

Analytics

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

Icon	Description	Icon	Description
	back		show minimum
	forward		show maximum
	forward to now		show line graph
	pause		show mountain graph
	zoom out		crop outliers
	zoom in		sync worksheet to this statistic
	show one minute		unsync worksheet statistics
	show one hour		drilldown

Icon	Description	Icon	Description
	show one day		export statistical data (download to client)
	show one week		save statistical data
	show one month		archive dataset
			send worksheet with support bundle

Identity Mapping

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

Icon*		Description	Icon*		Description
		allow Windows to Unix			allow Unix to Windows
		deny Windows to Unix			deny Unix to Windows
		allow bidirectional			

** Disabled icons shown at left.*

Miscellaneous Icons

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

Icon	Description	Icon	Description
	allow		SAS
	deny		SAS port
	storage pool		

CLI

CLI Introduction

The command line is an incredibly efficient and powerful tool for scripting repetitive administrative tasks. The appliance presents a command-line interface available through either the serial console, or SSH. There are several situations in which the preferred interaction with the system is command-line, as follows:

- Network unavailability - If the network is unavailable, browser-based management is impossible; the only vector for management is the serial 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

Logging Into the CLI

To log in remotely using the CLI, use an `ssh` client. If you have not configured other users 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:>
```

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
```

Contexts

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.

Root Context

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.

Child Contexts

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
dory:configuration preferences>
```

Dynamic Child Contexts

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
Bryan Cantrill          bmc                     31992       Dir
Super-User              root                    0           Loc
```

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

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

Returning to a Previous Context

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

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

Note that this will return to the previous context, which is not necessarily the parent context, as follows:

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

The done command can be used multiple times to backtrack to earlier contexts:

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

Navigating to a Parent Context

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

Contexts and Tab-Completion

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

Executing Context-Specific Commands

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
               session_timeout = 15
            session_annotation =
            advanced_analytics = false
```

If there is input following a command that changes context, that command will be executed in the target context, but control will return to the calling context. 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
```

Uncommitted Contexts

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 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 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
```

Properties

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
```

Getting Properties

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
```

Getting a Single Property Value

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
```

Tab Completion

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
```

Setting Properties

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 be "`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.

Committing a Set Property Value

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 with 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)
```

Setting a Property Value with an Implied Commit

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
dory:>
```

Setting a Property to a List of Values

Some properties take list of values. For these properties, the list elements should be separated by a comma. For example, NTP's `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
```

Setting a Property to a Value Containing Special Characters

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":

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

Immutable Properties

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:

```
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"
```

Scripting

Batching 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:

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

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

```
% 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:

```
% '''ssh root@dory << EOF
shares
select myproj
select myfs
snapshots snapshot newsnap
EOF'''
```

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.

Scripting

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

The Script Environment

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

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

As 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!
```

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)
```

Interacting with the System

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:

Function	Description
`get`	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.
`list`	Returns an array of tokens corresponding to the dynamic children of the current context.

Function	Description
run	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.
props	Returns an array of the property names for the current node.
set	Takes two string arguments, setting the specified property to the specified value.

The Run Function

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\n'
```

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+/))
[&#39;, 'boot_time', '=', '2009-10-12', '07:02:17', &#39;']
```

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

```
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\n',
           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': '');
```

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

```
% 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.

The List Function

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 ..');
               }

               run('cd ..');
       }
```

Here's 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
aleventhal/hw_eng 18432 266617007104
bcantrill/analytx 1073964032 266617007104
bgregg/dashbd 18432 266617007104
bgregg/filesys01 26112 107374156288
bpijewski/access_ctrl 18432 266617007104
...
```

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');

        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 = projects[i] + '/' + shares[j];
                        used = run('get space_data').split(/\s+/)[3];
                        avail = run('get space_available').split(/\s+/)[3];

                        printf('%-40s %-10s %-10s\n', share, used, avail);
                        run('cd ..');
                }

                run('cd ..');
        }
```

And here's 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
...
```

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. 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) {
                run(svcs[i]);
                if (props().length !== 0)
                        printf("%-10s %s\n", svcs[i], get('<status>'));
                run("done");
        }
.
```

Here's the output of running the script, assuming it were saved to a file named "svcinfo.aksh":

```
% ssh root@koi < space.aksh
Password:
cifs       disabled
dns        online
ftp        disabled
http       disabled
identity   online
idmap      online
ipmp       online
iscsi      online
ldap       disabled
ndmp       online
nfs        online
nis        online
ntp        online
scrk       online
sftp       disabled
smtp       online
snmp       disabled
ssh        online
tags       online
vscan      disabled
```

Generating Output

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

Function	Description
`dump`	Dumps the specified argument to the terminal, without expanding embedded newlines. Objects will be displayed in a JSON-like format. Useful for debugging.
`print`	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.
`printf`	Like C's printf(3C), prints the specified arguments according to the specified formatting string.

Dealing with Errors

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

- `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
```

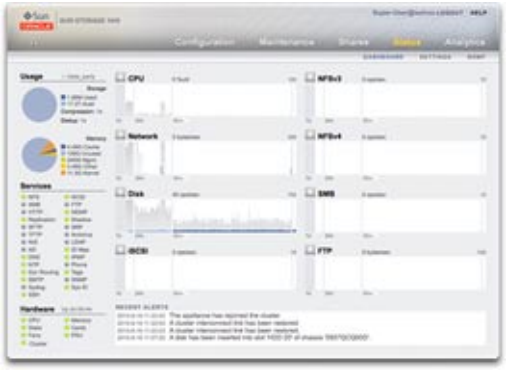
Automating Access

Whether using batched commands or scripting (or some combination), automated infrastructure requires automated access to the appliance. This should be done by creating users, giving them necessary authorizations, and uploading SSH keys.

◆ ◆ ◆ CHAPTER 2

Status

Status



Viewing the Sun Storage 7000 Status > Dashboard screen

Introduction

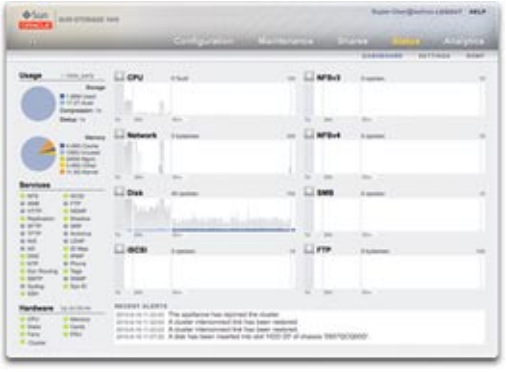
The Status section provides a summary of appliance status and configuration options. Refer to the following sections for conceptual and procedural information about appliance status views and related service configuration:

The Status > Dashboard screen provides a view of storage, memory, services, hardware, activity, and recent alerts.

The Status > Settings screen enables you to change the graphs that appear on the Dashboard and to customize the threshold settings associated with the weather icons shown for each graph on the Dashboard.

The Status > NDMP screen provides a view of any configured NDMP devices and recent activity for each NDMP session.

Dashboard



The Dashboard summarizes appliance status

Links

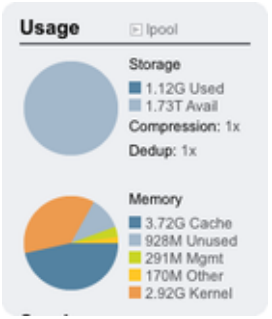
The Status Dashboard provides links to all the main screens of the browser user interface (BUI).

Over 100 visible items on the Dashboard link to associated BUI screens indicated by a boarder or highlighted text that appears on mouse-over.

The sections that follow describe the areas of the Dashboard in detail.

Usage

The Usage area of the Dashboard provides a summary of your storage pool and main memory usage. The pool name may be clicked to change the pool which is displayed on the status screen, should multiple pools be configured on the appliance.



Storage

The name of the pool appears at the top right of the Usage area. To the left is a pie-chart of used and available space. To go to the Shares screen for the pool, click the Storage pie-chart.

Memory

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.

Summary Pool Usage	
Used	Space used by this pool including data and snapshots.
Avail	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.
Compression	Current compression ratio achieved by this pool. Ratio will display 1x if compression is disabled.
Dedup	Current data deduplication ratio achieved by this pool. Ratio will display 1x if data deduplication is disabled.

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

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

Icons

Most services will be green indicating that the service is online, or grey indicating that the service is disabled.

See the icon status section for a reference of all possible states and icon colors.

Links

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.

Hardware

This area of the Dashboard shows an overview of hardware on the appliance.

Faults

If there is a known fault, the amber fault icon appears.

Links

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

Activity

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.

Graphs

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

Average

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.

Vertical Scale

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%.

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.

Weather

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 maintenance logs screen to examine all recent alerts in detail.

CLI

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

```
walu:> status dashboard
Storage:
   pool_0:
      Used     10.0G bytes
      Avail    6.52T bytes
      State          online
      Compression    1x

Memory:
   Cache        550M bytes
   Unused       121G bytes
   Mgmt         272M bytes
   Other       4.10G bytes
   Kernel      1.90G 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
   snmp              online                ssh               online
   tags              online                vscan             online

Hardware:
   CPU               online                Cards             online
   Disks             faulted               Fans              online
   Memory            online                PSU               online

Activity:
   CPU             1 %util                 Sunny
   Disk           32 ops/sec               Sunny
   iSCSI           0 ops/sec               Sunny
   NDMP            0 bytes/sec             Sunny
   NFSv3           0 ops/sec               Sunny
   NFSv4           0 ops/sec               Sunny
   Network       13K bytes/sec             Sunny
   SMB            0 ops/sec               Sunny

Recent Alerts:
   2009-10-13 07:46: A cluster interconnect link has been restored.
```

The previous descriptions in the BUI section apply, with the following differences:

- The activity plots aren't rendered in text (although we have thought about using aalib).
- The storage usage section will list details for all available pools in the CLI, whereas the BUI only has room to summarize one.

Separate views are available, for example `status activity show`:

```
caji:> status activity show
Activity:
   CPU            10 %util                 Sunny
   Disk          478 ops/sec               Partly Cloudy
   iSCSI           0 ops/sec               Sunny
   NDMP            0 bytes/sec             Sunny
   NFSv3         681 ops/sec               Partly Cloudy
   NFSv4           0 ops/sec               Sunny
   Network     22.8M bytes/sec             Partly Cloudy
   SMB             0 ops/sec              Sunny
caji:>
```

Tasks

Dashboard Tasks

▼ 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.

To reduce the browser memory growth (which will degrade image rendering performance), disable the memory cache in Firefox, as follows:

1. **Open about:config**
2. **Filter on "memory"**
3. **Set browser.cache.memory.enable = false**

Settings

Introduction

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.

BUI

Layout

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

Name	Units	Description
<empty>	-	No graph will be displayed in this location.
SMB	operations/sec	Average number of SMB operations.
CPU	utilization	Average cycles the appliance CPUs are busy. CPU cycles includes memory wait cycles.
Disk	operations/sec	Average number of operations to the physical storage devices.
HTTP	operations/sec	Average number of HTTP operations.
iSCSI	operations/sec	Average number of iSCSI operations.
Network	bytes/sec	Average bytes/sec across all physical network interfaces.
NDMP	bytes/sec	Average NDMP network bytes.
NFSv2	operations/sec	Average number of NFSv2 operations.
NFSv3	operations/sec	Average number of NFSv3 operations.
NFSv4	operations/sec	Average number of NFSv4 operations.
FTP	bytes/sec	Average number of FTP bytes.
SFTP	bytes/sec	Average number of SFTP bytes.

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

Thresholds

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.

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.

CLI

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.

Tasks

The following are examples tasks for this topic, with enumerated steps.

BUI

▼ 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

BUI

This page summarizes NDMP status, if the NDMP service has been configured and is active. Both backup devices and recent client activity are shown.

Devices

NDMP devices are listed here.

Field	Description	Examples
Type	Type of NDMP device	Robot, Tape drive
Path	Path of the NDMP device	`/dev/rmt/0n`
Vendor	Device vendor name	STK
Product	Device product name	SL500

Recent activity

This section summarizes recent NDMP activity.

Field	Description	Examples
ID	NDMP backup ID	49
Active	Backup currently active	No

Field	Description	Examples
Remote Client	NDMP client address and port	192.168.1.219:4760
Authenticated	Shows if the client has completed authentication yet	Yes, No
Data State	See Data State	Active, Idle, ...
Mover State	See Mover State	Active, Idle, ...
Current Operation	Current NDMP operation	Backup, Restore, None
Progress	A progress bar for this backup	

NDMP Data State

This field 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.

NDMP Mover State

This field shows the state of the NDMP device subsystem. Examples for tape devices:

- 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.

CLI

NDMP status is not currently available from the CLI.

◆ ◆ ◆ CHAPTER 3

Configuration

Configuration

The Configure Network screen

Introduction

This section describes how various properties of the appliance are configured, including network interfaces, services, and user accounts as follows.

- Initial - initial configuration
- Network - networking
- Services - data services
- SAN - storage area network configuration
- Cluster - clustering
- Users - user accounts and access control
- Preferences - user preferences
- Alerts - custom alerts
- Storage - reconfigure storage devices

For details about configuring or managing shares, see the Shares section. To backup the current configuration before making changes, use the Backup button on the Maintenance System screen.

Initial

Initial Configuration

The initial configuration consists of six configuration steps.

1. Network
2. DNS
3. Time
4. Name Services (NIS, LDAP, Active Directory)
5. Storage
6. Registration & Support

Prerequisites

The initial configuration of the system is conducted after powering it on for the first time and establishing a connection, as documented in the Installation section, also available as a PDF on http://wikis.sun.com/display/fishworks/Documentation. (`http://wikis.sun.com/display/fishworks/Documentation.`)

`Note` that the option to perform initial configuration of a cluster is only available in the BUI. If electing this option, read the clustering documentation included in the *Sun ZFS Storage 7000 Administration Guide* at http://wikis.sun.com/display/fishworks (`http://wikis.sun.com/display/fishworks`) before beginning initial configuration for detailed additional steps that are required for successful cluster setup. Pay careful attention to the Clustering Considerations for Networking section. 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.

Summary

This procedure will configure networking connectivity, several client network services, and the layout of the storage pool for standalone operation. When completed, the appliance is ready for use - but will not have any shares configured for remote clients to access. To create shares or revisit settings, refer to the Shares and Configuration sections of the Sun ZFS Storage 7000 System Administration Guide on http://wikis.sun.com/display/fishworks. (`http://wikis.sun.com/display/fishworks.`)

This procedure may be repeated at a later time by clicking the "INITIAL SETUP" button on the Maintenance > System screen or by entering the `maintenance system setup` context in the CLI.

BUI

The BUI initial configuration is the preferred method and provides a screen for each of the initial configuration steps.

Click Start to begin basic configuration of network, time directory and support services. Click Commit to save the configuration and go to the next screen. Arrows beneath the Commit button can be used to revisit previous steps, and change the configuration if desired.

Configuring Management Port

All standalone controllers should 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.

All cluster installations should 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.

CLI

Alternatively, use the CLI to step through 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.

Performing Initial Configuration with the CLI

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 the Network page for additional documentation or refer to the System Administration Guide at http://wikis.sun.com/display/fishworks. (`http://wikis.sun.com/display/fishworks.`) Pay careful attention to the Clustering Considerations for Networking section 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:

     DEVICE UP       MAC                     SPEED
       nge0 true     0:14:4f:8d:59:aa        1000 Mbit/s
       nge1 false    0:14:4f:8d:59:ab        0 Mbit/s
       nge2 false    0:14:4f:8d:59:ac        0 Mbit/s
       nge3 false    0:14:4f:8d:59:ad        0 Mbit/s

caji:maintenance system setup net> datalinks show
Datalinks:

   DATALINK CLASS          LINKS       LABEL
       nge0 device         nge0        Untitled Datalink

caji:maintenance system setup net> interfaces show
Interfaces:

  INTERFACE STATE  CLASS LINKS       ADDRS                 LABEL
       nge0 up     ip    nge0        192.168.2.80/22       Untitled Interface

caji:maintenance system setup net> done
```

Refer to the DNS section of the System Administration Guide at http://wikis.sun.com/display/fishworks (`http://wikis.sun.com/display/fishworks`) for additional documentation on 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 the NTP section of the System Administration Guide at http://wikis.sun.com/display/fishworks (http://wikis.sun.com/display/fishworks) 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" ...
```

Refer to the NIS, LDAP and Active Directory sections of the System Administration Guide at http://wikis.sun.com/display/fishworks (http://wikis.sun.com/display/fishworks) for additional documentation.

```
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. See the Storage section of the System Administration Guide at http://wikis.sun.com/display/fishworks (http://wikis.sun.com/display/fishworks) for additional documentation.

```
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" ...
```

Refer to (Phone Home) for additional documentation of remote support configuration.

```
Remote Support. Register your appliance and configure remote monitoring.

Subcommands that are valid in this context:

   tags                 => Configure service tags

   scrk                 => Configure phone home

   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 "support"

 caji:maintenance system setup support> done
 aksh: initial configuration complete!
```

Network

Configuring networking

Network Configuration

The Networking Configuration features permit you to create a variety of advanced networking setups out of your physical network ports, including link-aggregations, 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 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`** and **`VLAN`** 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.

In this 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.

To show this with an example, the following configuration is for a 4-way link aggregation:

Devices	Datalink	Interface
nge0, nge1, nge2, nge3	aggr1 (LACP aggregation)	deimos (192.168.2.80/22)

The datalink entity (which we named "aggr1") groups the network devices in a configurable way (LACP aggregation policy). The interface entity (which we named "deimos") provides configurable IP address settings, which it makes available on the network via the datalink. The network devices (named "nge0", "nge1", ..., 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. An example of a single IP address on a single port (common configuration) is:

Devices	Datalink	Interface
nge0	datalink1	deimos (192.168.2.80/22)

Devices

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

Datalinks

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

- VLANs - Virtual LANs to improve local network security and isolation.
- LACP - Link Aggregation Control Protocol, to bundle multiple network devices to behave as one. This improves performance (multiplies bandwidth) and reliability (can survive network port failure), however the appliance must be connected to a switch that supports LACP and has it enabled for those ports.
- IB Partitions - Infiniband partitions to connect to logically isolated IB fabric domains.

The following datalink settings are available:

Property	Description
Name	Custom name for the datalink. For example: "internal", "external", "adminnet", etc.
VLAN	Use VLAN headers
VLAN ID	VLAN ID
Jumbo Frames	Use a large MTU (~9000 bytes, depending on the hardware and device driver), to improve network performance. Successful use of this option requires that attached switches support this feature. Once the Jumbo Frames option is enabled 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.
LACP Aggregation	Aggregate multiple network devices

Property	Description
LACP Policy	Policy for picking outbound port. L2 hashes the source and destination MAC address; L3 uses the source and destination IP address; L4 uses the source and destination transport level port
LACP Mode	Switch 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.
LACP Timer	For Active mode, this is the interval between LACP messages
IB Partition	Use IB partitions
Partition Key	This property designates 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 IPMP and clustering rules on the sub-net manager.

Interfaces

These configure IP addresses via datalinks. They support:

- 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:

Property	Description
Name	Custom name for the interface
Allow Administration	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
Enable Interface	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.
IPv4 Configure with	Either "Static Address List" manually entered, or "DHCP" for dynamically requested
IPv4 Address/Mask	One or more IPv4 addresses in CIDR notation (192.168.1.1/24)

Property	Description
IPv6 Configure with	Either "Static Address List" manually entered, or "IPv6 AutoConfiguration" to use automatically generated link-local address (and site-local if an IPv6 router responds)
IPv6 Address/Mask	One or more IPv6 addresses in CIDR notation (1080::8:800:200C:417A/32)
IP MultiPathing Group	Configure IP multipathing, where a pool of datalinks can be used for redundancy

IP MultiPathing (IPMP)

IP MultiPathing groups are used to provide IP addresses that will remain available in the event of a IP interface failure (such 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.

If probe-based failure detection is enabled on an IP interface (i.e., a test address is configured), 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 224.0.01. 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.

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).

Step by step instructions for building IPMP groups can be found in the Tasks section below.

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.

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.

The two technologies complement each other and can be deployed together to provide the combined benefits of network performance and availability.

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.

Routing Entries

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

Field	Description	Examples
Destination	Range of IP destination addresses (in CIDR notation) that can match the route	192.168.0.0/22
Gateway	Next hop (IP address) to send the packet to (except for "system" routes -- see below)	192.168.2.80
Family	Internet protocol	IPv4, IPv6
Type	Origin of the route	dhcp, static, system
Interface	IP interface the packet will be sent on	nge0

A routing entry with a "destination" field of `0.0.0.0/0` will match 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:

Kind	Route kind	Default, Network

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:

Type	Description
Static	Created and managed by the appliance administrator.
System	Created automatically by the appliance as part of enabling an IP interface. A system route will be created for each IP subnet the appliance can directly reach. Since these routes are directly reachable, the "gateway" field instead identifies the appliance's IP address on that subnet.
DHCP	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.
Dynamic	Created automatically by the appliance via the RIP and RIPng dynamic routing protocols (if enabled).

One additional type identifies a static route that cannot currently be used:

Inactive	Previously-created static route associated with a disabled or offline IP interface.

Routing Properties

Property	Description
Multihoming model	Controls the system policy for accepting and transmitting IP packets when multiple IP interfaces are simultaneously enabled. Allowed values are "loose" (default), "adaptive", and "strict". See the discussion below.

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:

Policy	Description
Loose	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.

Policy	Description
Adaptive	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.
Strict	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 will ignore any routes that have gateway addresses on a different subnet from the packet's source address. If no eligible routes remain, drop the packet.

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.

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:

icon	description
	Add new datalink/interface/route

icon	description
	Edit datalink/interface/route settings
	Destroy datalink/interface/route
	Drag-and-drop icon
	connected network port
	connected network port with I/O activity
	disconnected network port (link down, cable problem?)
	active Infiniband port
	active Infiniband port with I/O activity
	inactive Infiniband port (down, init, or arm state)
	Infiniband partition device is up
	Infiniband partition device is down (subnet manager problem)
	network datalink
	network datalink VLAN
	network datalink aggregation
	network datalink aggregation VLAN
	network datalink IB partition
	interface is being used to send and receive packets (either up or degraded)
	interface has been disabled by the user
	interface is offline (owned by the cluster peer)
	interface has failed or has been configured with a duplicate IP address

At top right is local navigation for Configuration, Addresses and Routing, which display alternate configuration views.

Configuration

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. Other moves are possible. This can be helpful for complex configurations, where valid moves are highlighted.

Addresses

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

Field	Description	Example
Network Datalink	Datalink name and detail summary	datalink1 (via nge0)
Network Interface	Interface name and details summary	IPv4 DHCP, via datalink1
Network Addresses	Addresses hosted by this interface	192.168.2.80/22
Host Names	Resolved host names for the network addresses	caji.sf.example.com

Routing

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.

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:

DEVICE      UP     SPEED         MAC
nge0        true   1000 Mbit/s   0:14:4f:9a:b9:0
nge1        true   1000 Mbit/s   0:14:4f:9a:b9:1
nge2        true   1000 Mbit/s   0:14:4f:9a:b8:fe
nge3        true   1000 Mbit/s   0:14:4f:9a:b8:ff

caji:configuration net> datalinks show
Datalinks:

   DATALINK CLASS          LINKS       LABEL
       nge0 device         nge0        datalink1

caji:configuration net> interfaces show
Interfaces:

  INTERFACE STATE  CLASS LINKS       ADDRS                LABEL
       nge0 up     ip    nge0        192.168.2.80/22      caji

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

Routes:

ROUTE      DESTINATION                    GATEWAY         INTERFACE TYPE
route-000  0.0.0.0/0                      192.168.1.1     nge0      dhcp
route-001  192.168.0.0/22                 192.168.2.142   nge0      system
```

Type `help` in each section to see the relevant commands for creating and configuring datalinks, interfaces, and routes.

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=nge1
                         links = nge1 (uncommitted)
caji:configuration net datalinks device (uncommitted)> set label=datalink2
                         label = datalink2 (uncommitted)
caji:configuration net datalinks device (uncommitted)> set jumbo=true
                         jumbo = true (uncommitted)
caji:configuration net datalinks device (uncommitted)> commit
caji:configuration net datalinks> show
Datalinks:

   DATALINK CLASS          LINKS       LABEL
       nge0 device         nge0        datalink1
       nge1 device         nge1        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=nge1
                        links = nge1 (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
caji:configuration net interfaces> show
Interfaces:

  INTERFACE STATE  CLASS LINKS       ADDRS                LABEL
       nge0 up     ip    nge0        192.168.2.80/22      caji
       nge1 up     ip    nge1        10.0.1.1/8           caji2
```

The following demonstrates creating a default route via `10.0.1.2` over the new `nge1` IP interface:

```
caji:configuration net routing> create
caji:configuration net route (uncommitted)> set family=IPv4
                    family = IPv4 (uncommitted)
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=nge1
                    interface = nge1 (uncommitted)
caji:configuration net route (uncommitted)> set gateway=10.0.1.2
                    gateway = 10.0.1.2 (uncommitted)
caji:configuration net route (uncommitted)> commit
```

Tasks

BUI

▼ Creating a single port interface

1. Click the Datalink ⊕ icon.
2. Optionally set name and jumbo frames.
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

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, drag-and-drop

1 Mouse over a device and click the drag-and-drop icon ( ).

2 Drag it to the Datalink list and release.

3 Optionally set name and jumbo frames.

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

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".

5 Click the Interfaces icon.

6 Set desired properties, choose the aggregated link from the Datalinks list, and click "APPLY".

7 Click "APPLY" at the top to commit the configuration.

▼ Create an IPMP group using probe-based and link-state failure detection

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 fist step from the Interfaces list.

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

8 Click "APPLY".

▼ Create 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".

Extend an LACP aggregation

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.

Extend an IPMP group

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.

Create an Infiniband partition datalink and interface

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. Click "APPLY". The new partition datalink will appear in the Datalinks list.
6. Click the Interface icon.
7. Set desired properties, and choose the datalink previously created.
8. Click "APPLY". The interface will appear in the Interfaces list.
9. The running appliance network configuration has not yet changed. When you are finished configuring interfaces, click "APPLY" at the top to commit the configuration.

Adding a static route

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

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

CLI

▼ Adding a static route

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

▼ Deleting a static route

1. Go to `configuration network routing`.
2. Type `show` to list routes, and route names (eg, `route-002`).
3. Enter `destroy` *route name*.

Infiniband Upgrade Procedures for Q3.2010

The administrative model for Infiniband datalink partitions has changed. Infiniband datalinks and interfaces will not be preserved across an upgrade to Q3.2010. You must teardown all Infiniband-based interfaces and datalink (partitions) prior to beginning the upgrade to Q3.2010. Once the system has been upgraded, the Infiniband datalink partitions and interfaces may be re-created. There are no actions required on the subnet manager or switch. Failure to tear down the interfaces and datalinks will result in sfaulted datalink and non-functioning interfaces.

Storage

Introduction

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 the Shares section.

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 a Sun Storage 7000 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 overprovisioning by approximately 20%. The Shares UI can be used to determine how much space is currently being used.

Configure

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.

After the task is started, storage configuration falls into two different phases: verification and configuration.

Verification and Allocation

The verification phase allows you to verify that all storage is attached and functioning, and allocate disks within chassis. In a standalone system, this presents a list of all available storage

and drive types, with the ability to change the number of disks to allocate to the new pool. By default, the maximum number of disks are allocated, but this number can be reduced in anticipation of creating multiple pools.

In an expandable system, JBODs are displayed in a list along with the head node, and allocation can be controlled within each JBOD. For each JBOD, the system must import available disks, a process that can take a significant amount of time depending the number and configuration of JBODs. Disks within the system chassis can be allocated individually (as with cache devices), but JBODs must be allocated as either 'whole' or 'half'. In general, whole JBODs are the preferred unit for managing storage, but half JBODs can be used where storage needs are small, or where NSPF is needed in a smaller configuration.

Attempting to commit this step using chassis with missing or failed devices will result in a warning. Once you configure a storage pool in this manner, you will never be able to add the missing or broken disk. Therefore it is important that **`all devices must be connected and functioning`** before continuing past the verification step.

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.

Data Profile	Description
Double parity RAID	RAID in which each stripe contains two parity disks. This yields high capacity and high availability, as data remains available even with the failure of any two disks. The capacity and availability come at some cost to performance: parity needs to be calculated on writes (costing both CPU and I/O bandwidth) and many concurrent I/Os need to be performed to access a single block (reducing available I/O operations). The performance effects on read operations are often greatly diminished when cache is available.
Mirrored	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).

Data Profile	Description
Single parity RAID, narrow stripes	RAID in which each stripe is kept to three data disks and a single parity disk. At normal stripe widths, single parity RAID offers few advantages over double parity RAID -- and has the major disadvantage of only being able to survive a single disk failure. However, at narrow stripe widths, this single parity RAID configuration can fill a gap between mirroring and double parity RAID: its narrow width offers better random read performance than the wider stripe double parity configuration, but it does not have quite the capacity cost of a mirrored configuration. While this configuration may be an appropriate compromise in some situations, it is generally not recommended unless capacity and random read performance must be carefully balanced: those who need more capacity are encouraged to opt for a wider, double-parity configuration; those for whom random read performance is of paramount importance are encouraged to consider either a mirrored configuration or (if the workload is amenable to it) a double parity RAID configuration with sufficient memory and dedicated cache devices to service the workload without requiring disk-based I/O.
Striped	Data is striped across disks, with no redundancy whatsoever. While this maximizes both performance and capacity, it comes at great cost: a single disk failure will result in data loss. This configuration is not recommended, and should only be used when data loss is considered to be an acceptable trade off for marginal gains in capacity and performance.
Triple parity RAID, wide stripes	RAID in which each stripe has three disks for parity, and for which wide stripes are configured to maximize for capacity. Wide stripes can exacerbate the performance effects of double parity RAID: while bandwidth will be acceptable, the number of I/O operations that the entire system can perform will be greatly diminished. 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.
Triple mirrored	Data is triply mirrored, reducing capacity by one third, but yielding a very highly reliable and high-performing system. This configuration is intended for situations in which maximum performance, and availability are required while capacity is much less important (for example, database storage). Compared with a two-way mirror, a three-way mirror adds additional protection against disk failures and latent disk failures in particular during reconstruction for a previous failure.

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 JBOD failure will not result in data loss. Note that systems are already configured with redundancy across nearly all components. Each JBOD 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 JBOD). In general, adopting NSPF will result in lower capacity, as it has more stringent requirements on stripe width.

Log devices can also have one of two different profiles: striped or mirrored. The data on log devices is only used in the event of node failure, so in order to lose data with an unmirrored log device it is necessary for both the device to fail and the node to reboot within a few seconds. This constitutes a double failure, but using mirrored log devices can make this effectively impossible, requiring two simultaneous device failures and node failure within a very small time window.

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 a double parity RAID configuration with wide stripes. This has been supplanted by the triple parity RAID, wide stripe configuration as it adds significantly better reliability. Pools configured with double parity RAID with wide stripes under a previous software version continue to be supported but newly configured or reconfigured pools cannot select that option.

Import

This allows you to import an existing storage pool, as well as any inadvertently unconfigured pools. This can be used after a 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 'import' command.

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.

Add

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 JBOD to a double parity RAID-Z NSPF config makes it impossible to preserve NSPF characteristics. However, you can still add the JBOD and create RAID stripes within the JBOD, sacrificing NSPF in the process.

Unconfig

This will remove 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.

Scrub

This will initiate the storage pool scrub process, which will verify all content to check for errors. If any unrecoverable errors are found, either through a scrub or through normal operation, the BUI will display the affected files. The scrub can also be stopped if necessary.

Tasks

BUI

▼ Configuring a Storage Pool

There are three different 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 JBOD allocation for the storage pool. JBOD allocation may be none, half or all. If no JBODs are detected, check your JBOD 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".**

SAN

SAN

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

- A client which will access the storage on the network
- A storage appliance which will provide the storage on the network
- A network to 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.

Terminology

To configure the appliance to operate on a SAN, it is essential to understand some basic terms:

Term	Description
Logical Unit	A term used to describe a component in a storage system. Uniquely numbered, this creates what is referred to as a Logicial 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.
Target	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's administrator, and is identified by unique addressing methods. A target, once configured, consists of zero or more logical units.
Target group	A set of targets. LUNs are exported over all the targets in one specific target group.
Initiator	An application or production system end-point that is capable of initiating a SCSI session, sending SCSI commands and I/O requests. Initiators are also identified by unique addressing methods.
Initiator group	A set of initiators. When an initiator group is associated with a LUN, only initiators from that group may access the LUN.

Additionally, it is important to be aware of SCSI transport protocols:

Term	Description
FC	Fibre Channel, a protocol for sharing SCSI based storage over a storage area network (SAN), consisting of fiber-optic cables, FC switches and HBAs.
iSCSI	Internet SCSI, a protocol for sharing SCSI based storage over IP networks.
iSER	iSCSI Extension for RDMA, a protocol that maps the iSCSI protocol over a network that provides RDMA services (i.e. InfiniBand). The iSER protocol is transparently selected by the iSCSI subsystem, based on the presence of correctly configured IB hardware. In the CLI and BUI, all iSER-capable components (targets and initiators) are managed as iSCSI components.
SRP	SCSI RDMA Protocol, a protocol for sharing SCSI based storage over a network that provides RDMA services (i.e. InfiniBand).

Targets and Initiators

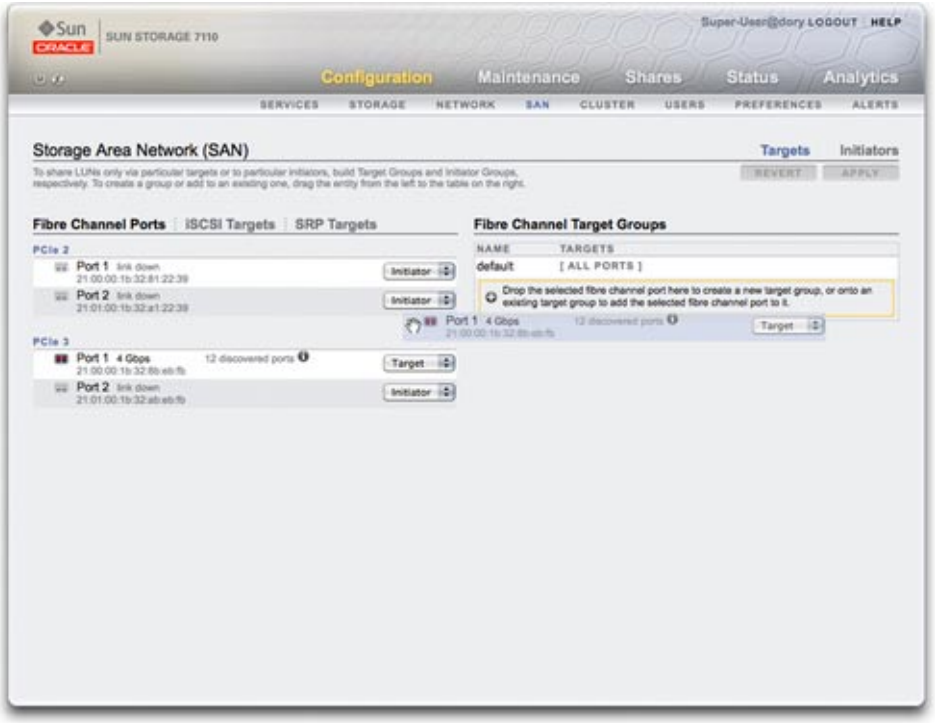
Targets and initiators are configured by protocol. Refer to the documentation on a particular protocol (FC, iSCSI or SRP) for details.

Target and Initiator Groups

Target and initiators 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 that is associated with an initiator group can only be seen by the initiators in the group. 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.

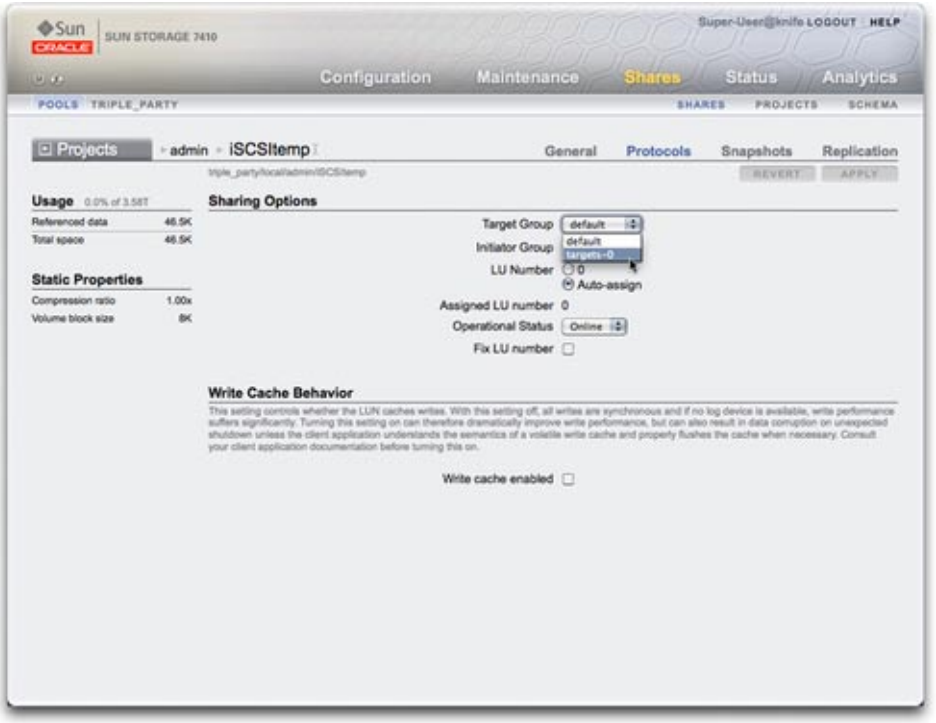
BUI

The following screenshot shows the Configuration > SAN screen. Use the Target and Initiator links to navigate. Then, click the Fibre Channel Ports, iSCSI Targets, or SRP Targets links to operate on targets by protocol type.



The above screenshot demonstrates creating a new FC target group by moving a discovered target to the Fibre Channel target list.

The following screenshot shows the Shares > Shares > Protocols screen. Use the Target Group or Initiator Group menus to associate a LUN.



CLI

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.

Terms

SAN Terminology

The purpose of this section is to familiarize the reader with the various concepts involved in configuring a SAN (storage area network). Before getting into the details of how to configure and manage block storage on the appliance, it would be useful to understand some terms.

Term	Description
SCSI Target	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.
SCSI Initiator	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).
Logical Unit	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 Logicial 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.
iSCSI	Internet SCSI, a protocol for sharing SCSI based storage over IP networks.
iSER	iSCSI Extension for RDMA, a protocol that maps the iSCSI protocol over a network that provides RDMA services (i.e. InfiniBand). The iSER protocol is transparently selected by the iSCSI subsystem, based on the presence of correctly configured IB hardware. In the CLI and BUI, all iSER-capable components (targets and initiators) are managed as iSCSI components.
FC	Fibre Channel, a protocol for sharing SCSI based storage over a storage area network (SAN), consisting of fiber-optic cables, FC switches and HBAs.
SRP	SCSI RDMA Protocol, a protocol for sharing SCSI based storage over a network that provides RDMA services (i.e. InfiniBand).
IQN	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, "com.sun". Everything following is a unique ID that Sun uses to identify the target.
Target portal	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.
Target portal group	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.
CHAP	Challenge-handshake authentication protocol, a security protocol which can authenticate a target to an initiator, an initiator to a target, or both.
RADIUS	A system for using a centralized server to perform CHAP authentication on behalf of storage nodes.

Term	Description
Target group	A set of targets. LUNs are exported over all the targets in one specific target group.
Initiator group	A set of initiators. When an initiator group is associated with a LUN, only initiators from that group may access the LUN.

Each LUN has several properties which control how the volume is exported. See the Protocols section for more information.

FC

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.

Target Configuration

By default, all FC ports are used by the appliance to connect to a tape SAN for purposes of backup; to enable sharing via FC, one or more FC ports must be configured to be in *target mode*. Configuring a port to be in target mode requires the appliance to be reset, but multiple ports may be configured to be in target mode simultaneously. The mode of a port will be preserved across any subsequent reboots or upgrades. You must have `root` permissions to change the mode of a port.

Each FC port is assigned a World Wide Name (WWN), and -- as with other block protocols -- FC targets may be grouped into target 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.

Clustering Considerations

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) are *active*; the second path (or set of paths) are *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.

Initiator Configuration

Initiators are identified by their WWN, and 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. Further, and 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.

Switch Considerations

In general, the FC target works with off-the-shelf FC switches. In particular, the following switches have been tested and are known to work:

Switch vendor	Model
Brocade	All models supporting speeds of 4 Gb/sec and/or 8 Gb/sec
Cisco	All models supporting speeds of 4 Gb/sec and/or 8 Gb/sec
Qlogic	All models supporting speeds of 4 Gb/sec and/or 8 Gb/sec.

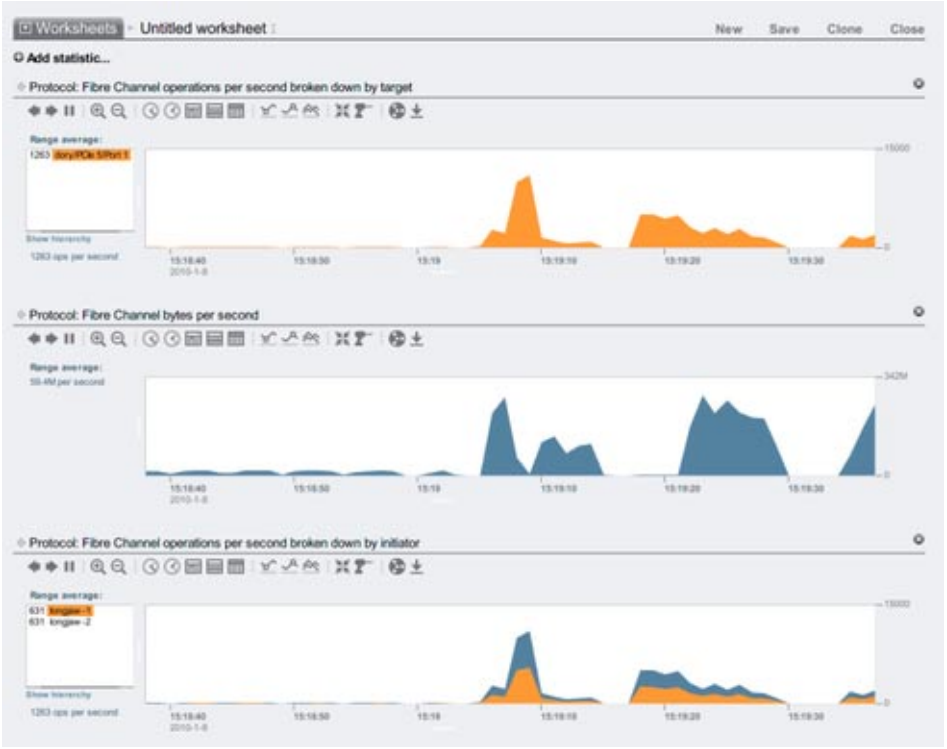
Clustering Considerations

As discussed in target clustering considerations, 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. The following initiators have been tested and are known to work in a cluster:

Operating system	HBA	Additional information
OpenSolaris 2010.03	QLogic, Emulex	Configuration notes
Solaris 10 10/09	QLogic, Emulex	Configuration notes
Windows 2008	QLogic, Emulex	Configuration notes
Red Hat Enterprise Linux 5.4	QLogic, Emulex	Configuration notes
Oracle Enterprise Linux 5.4	QLogic, Emulex	Configuration notes
VMware ESX 4.0	QLogic, Emulex	Configuration notes

Performance Considerations

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



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.

Troubleshooting

Queue Overruns

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 7410 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`.

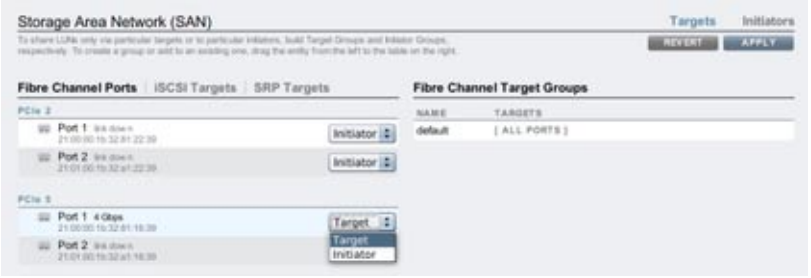
Link-level Issues

To troubleshoot link-level issues such as broken or flakey 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.

BUI

Changing modes of FC ports

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, see the clustering considerations section.

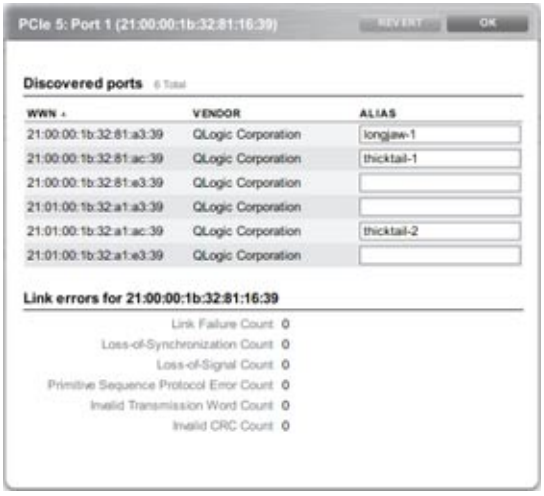


After setting desired ports to Target, click the Apply button. A confirmation message will appear notifying you that the appliance will reboot immediately. Confirm that you want to reboot.

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

Viewing discovered FC ports

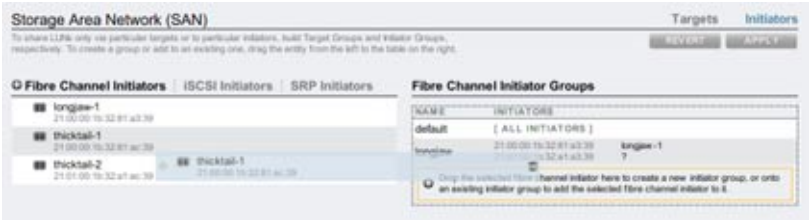
Click the info icon to view the Discovered Ports dialog where you can troubleshoot link errors. In the Discovered Ports dialog, click a WWN in the list to view associated link errors.



Creating FC Initiator Groups

Create and manage initiator groups on the Initiators screen. Click the add icon to view unaliased ports. Click a WWN in the list to add a meaningful alias in the Alias field.

On the Initiators screen, drag initiators to the FC Initiator Groups list to create new groups or add to existing groups.



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

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.

CLI

Changing modes of FC ports

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

Viewing discovered FC ports

```
dory:configuration san targets fc> show
Properties:
                     targets = wwn.2100001B32811639,wwn.2101001B32A12239
```

```
Targets:
NAME       MODE       WWN                      PORT                 SPEED
target-000 target     wwn.2100001B32811639     PCIe 5: Port 1       4 Gbit/s
target-001 initiator  wwn.2101001B32A11639     PCIe 5: Port 2       0 Gbit/s
target-002 initiator  wwn.2100001B32812239     PCIe 2: Port 1       0 Gbit/s
target-003 target     wwn.2101001B32A12239     PCIe 2: Port 2       0 Gbit/s
dory:configuration san targets fc> select target-000
dory:configuration san targets fc target-000> show
Properties:
                          wwn = wwn.2100001B32811639
                         port = PCIe 5: Port 1
                         mode = target
                        speed = 4 Gbit/s
              discovered_ports = 6
            link_failure_count = 0
            loss_of_sync_count = 0
          loss_of_signal_count = 0
          protocol_error_count = 0
         invalid_tx_word_count = 0
             invalid_crc_count = 0
Ports:
PORT      WWN                   ALIAS              MANUFACTURER
port-000  wwn.2100001B3281A339  longjaw-1          QLogic Corporation
port-001  wwn.2101001B32A1A339  longjaw-2          QLogic Corporation
port-002  wwn.2100001B3281AC39  thicktail-1        QLogic Corporation
port-003  wwn.2101001B32A1AC39  thicktail-2        QLogic Corporation
port-004  wwn.2100001B3281E339  <none>             QLogic Corporation
port-005  wwn.2101001B32A1E339  <none>             QLogic Corporation
```

Creating FC Initiator Groups

```
dory:configuration san initiators fc groups> create
dory:configuration san initiators fc group (uncommitted)> set name=lefteye
dory:configuration san initiators fc group (uncommitted)>
    set initiators=wwn.2101001B32A1AC39,wwn.2100001B3281AC39
dory:configuration san initiators fc group (uncommitted)> commit
dory:configuration san initiators fc groups> list
GROUP     NAME
group-001 lefteye
          |
          +-> INITIATORS
              wwn.2101001B32A1AC39
              wwn.2100001B3281AC39
```

Associating a LUN with an FC initiator group

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

```
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

Refer to the CLI Usage and Simple CLI Scripting and Batching Commands sections for information about how to modify and use the following example script.

```
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'
               },
               tecopa: {
                       'tecopa-1': 'wwn.2100001b3281e339',
                       'tecopa-2': 'wwn.2101001b32a1e339'
               },
               spinedace: {
                       'spinedace-1': 'wwn.2100001b3281df39',
                       'spinedace-2': 'wwn.2101001b32a1df39'
               },
               fera: {
                       'fera-1': 'wwn.2100001b32817939',
                       'fera-2': 'wwn.2101001b32a17939'
               }
       };
       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)\n');
                               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\n');
                       }
                       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)\n');
                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\n');
        }
        run('cd ..');
}
```

FCMPxIO

Configuring FC Client Multipathing

The Sun ZFS Storage 7000 series uses Asymmetric Logical Unit Access (ALUA) to provide FC target multipathing support. Please refer to SCSI Primary Commands (SPC) definition on t10 at http://www.t10.org (`http://www.t10.org`) if you need more information.

The following instructions provide a guide for setting up the FC host clients that are connected to a FC target enabled clustered appliance.

Configuring Solaris Initiators

FC target on a clustered appliance was qualified with OpenSolaris 2010.03 and Solaris 10 10/09. It is recommended that users with earlier versions of Solaris 10 on their clients upgrade to 10/09 or later for FC connectivity to a clustered appliance. If using Solaris 10 10/09, users must also apply latest MPxIO patch: 143120-03 (Sparc), 143121-03 (x86).

MPxIO is enabled on Solaris x86 platforms but disabled on SPARC by default. The `mpathadm show LU` command shows the path state changing from active to standby or standby to active. Alternately, you can also use `luxadm display` to show path state.

The `stmsboot` utility enables and disables MPxIO, for example:

1. To enable MPxIO, run `stmsboot -D fp -e`
2. To disable MPxIO, run `stmsboot -D fp -d`
3. To verify the state, run `mpathadm show LU`

Configuring Windows Initiators

ALUA multipathing is supported by native Windows 2008/R2 MPIO only.

1. Verify that the FC HBA Windows driver is installed and the HBA is operational.
2. Install or verify installation of the Windows Server 2008 MPIO Optional Component. Configure multipath support for the SS7000 by issuing the `mpclaim.exe -r -i -a ""`command at a Windows Command Prompt. This will force a system reboot and is necessary to complete MPIO setup and ensure proper path/LUN discovery.
3. Once the client has rebooted, verify that Windows Client can discover and access SS7000 LUN(s) and the correct number of paths and path states are displayed. This can be verified using the Windows Disk Management utility. For each LUN on the SS7000 there should be only one corresponding disk available in the Disk Management GUI.
4. In the event of a SS7000 node failure, the default Microsoft DSM timer counters may be insufficient to ensure I/O continues uninterrupted. To alleviate this, we recommend setting the following Timer Counter values in the DSM details section of a disks Multi-Path Disk Device properties.

Windows Tunables - Microsoft DSM Details

Windows Tunable	Description	Default Value	Recommended Value
PathVerifyEnabled	Enables path verification by MPIO on all paths every âNâ seconds . N depends on the value set in PathVerificationPeriod.	Disabled	Enabled
PathVerificationPeriod	Used to indicate the periodicity (in seconds) with which MPIO has been requested to perform path verification. This field is only used if PathVerifyEnabled = TRUE.	30 seconds	5 seconds
RetryInterval	Specifies the interval of time after which a failed request is retried (after the DSM has decided so, and assuming that the IO has been retried less number of times than RetryCount).	1 second	5 seconds
RetryCount	Specifies the number of times a failed IO occurs before the DSM determines that a failing request must be retried.	3	300
PDORemovePeriod	Controls the amount of time (in seconds) that the multipath pseudo-LUN will continue to remain in system memory, even after losing all paths to the device.	20 seconds	1500 seconds

Errata:

- **Emulex HBAs and Windows Server 2008:** When using a Windows Server 2008 client equipped with Emulex HBAs, a change to the HBA driver parameter is required. In order to ensure uninterrupted I/O during a cluster failover/failback operation, you must modify the Emulex HBA NodeTimeout value, setting it to 0. Use the Emulex *OCManager Utility*, available from http://www.emulex.com (`http://www.emulex.com`) to adjust this parameter.

Configuring Linux Initiators

The following instructions cover Red Hat Enterprise Linux 5.4 (RHEL 5.4) and Oracle Enterprise Linux 5.4 (OEL 5.4).

1. Ensure the correct device-mappers are installed.

2. Stop the `multipathd` service.

```
# service multipathd stop
Stopping multipathd daemon:                                [  OK  ]
```

3. Modify `/etc/multipath.conf` to enable SUN arrays by adding the following lines under the devices sections.

```
device
{
vendor            "SUN"
product            "Sun Storage 7310" or
                       "Sun Storage 7410" (depending on storage system)
getuid_callout         "/sbin/scsi_id -g -u -s /block/%n"
prio_callout             "/sbin/mpath_prio_alua /dev/%n"
hardware_handler    "0"
path_grouping_policy     group_by_prio
failback         immediate
no_path_retry             queue
rr_min_io         100
path_checker         tur
rr_weight         uniform
}
```

4. Enable multipath and verify by starting the `multipathd` service.

```
#service multipathd start
Starting multipathd daemon:                                [  OK  ]
```

5. Run the `multipath` command after the SCSI bus rescan is finished to verify multipath I/O is enabled. Note that standby paths will be shown as due to a known Linux bug. For this reason, it is recommended that users verify the paths are actually operational before putting the system into production. For more details, refer to the Troubleshooting section below.

```
#multipath âll
sdd: checker msg is "tur checker reports path is down"
mpath1 (3600144f094f0bd0300004b31c88f0001) dm-2 SUN,Sun Storage 7410 (or 7310)
[size=20G][features=1 queue_if_no_path][hwhandler=0][rw]
\_ round-robin 0 [prio=50][active]
\_ 2:0:0:0 sdb 8:16  [active][ready]
\_ round-robin 0 [prio=0][enabled]
\_ 2:0:1:0 sdd 8:48  [failed][faulty]
```

Configuring VMware ESX Initiators

1. Verify the current SATP plugin that is in use by issuing the `esx nmp device list` command

```
# esxcli nmp device list
naa.600144f0ed81720500004bb3c1f60002
   Device Display Name: SUN Fibre Channel Disk (naa.600144f0ed81720500004bb3c1f60002)
   Storage Array Type: VMW_SATP_DEFAULT_AA
   Storage Array Type Device Config:
   Path Selection Policy: VMW_PSP_FIXED
   Path Selection Policy Device Config: {preferred=vmhba0:C0:T1:L0;current=vmhba0:C0:T1:L0}
   Working Paths: vmhba0:C0:T1:L0
```

VMW_SATP_DEFAULT_AA is the default plugin. This plugin is not ALUA-capable.

2. Verify the correct Vendor and Model string for the specific Sun Storage 7000 with the `dmesg` command.

```
# dmesg | grep SUN
[   29.826974]   Vendor: SUN       Model: Sun Storage 7410  Rev: 1.0
```

3. Add a rule to enable the ALUA plugin for the Sun Storage 7000 by using the `esxcli nmp satp addrule` command.

```
# esxcli nmp satp addrule -s VMW_SATP_ALUA -e "Sun Storage 7000" -V "SUN" -M "Sun Storage 7410" -c "tpgs_on"
```

or

```
# esxcli nmp satp addrule -s VMW_SATP_ALUA -e "Sun Storage 7000" -V "SUN" -M "Sun Storage 7310" -c "tpgs_on"
```

4. Verify the rule was correctly added.

```
# esxcli nmp satp listrules | grep SUN
VMW_SATP_ALUA        SUN      Sun Storage 7410     tpgs_on        Sun Storage 7000 Cluster
```

5. Reboot the VMware ESX server. When server has rebooted, check to ensure the correct plugin is now in effect with the `esxcli nmp device list` command.

```
# esxcli nmp device list
naa.600144f0ed81720500004bb3c1f60002
   Device Display Name: SUN Fibre Channel Disk (naa.600144f0ed81720500004bb3c1f60002)
   Storage Array Type: VMW_SATP_ALUA
   Storage Array Type Device Config: {implicit_support=on;explicit_support=off;
   explicit_allow=on;alua_followover=on;{TPG_id=0,TPG_state=STBY}{TPG_id=1,TPG_state=AO}}
   Path Selection Policy: VMW_PSP_MRU
   Path Selection Policy Device Config: Current Path=vmhba1:C0:T1:L0
   Working Paths: vmhba1:C0:T1:L0
```

Troubleshooting

This section describes troubleshooting known issues.

Multipath-tools version 0.4.7 bundled in RHEL 5.4 and OEL 5.4 is unable to recognize paths in ALUA standby access state

In SCSI spec, a target port which is in standby state does not respond to Test Unit Ready command, so standby paths are shown as in multipath command output.

The fix for this problem is committed into the multipath-tool source tree on 2009-04-21 (which is later than its 0.4.8 official release). Users have to obtain the latest version of the multipath-tool source code from: http://christophe.varoqui.free.fr/ (http://christophe.varoqui.free.fr/)

Users should get the latest source code from its git repository. The multipath-tools-0.4.8.tar.bz2 tarball does not contain the fix.

Finally, the status shown in multipath command output does not impact functionalities like I/O and failover/failback, so updating the package is not mandatory.

See Also

- fcinfo man page

http://docs.sun.com/app/docs/doc/816-5166/fcinfo-1m (http://docs.sun.com/app/docs/doc/816-5166/fcinfo-1m)?l=en&a=view&q=fcinfo

- Solaris Fibre Channel and Storage Multipathing Administration Guide

http://docs.sun.com/source/819-0139-12/ (http://docs.sun.com/source/819-0139-12/)

- Windows Server High Availability with Microsoft MPIO

http://www.microsoft.com/downloads/details.aspx (http://www.microsoft.com/downloads/details.aspx)?FamilyID=CBD27A84-23A1-4E88-B198-6233623582F3&displaylang=en

- Using Device-Mapper Multipath - Red Hat

http://www.redhat.com/docs/manuals/csgfs/browse/4.6/DM_Multipath/index.html (http://www.redhat.com/docs/manuals/csgfs/browse/4.6/DM_Multipath/index.html)

iSCSI

Introduction

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

Target Configuration

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.

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 Sun uses to identify the target.

Target Property	Description
Target IQN	The IQN for this target. The IQN can be manually specified or auto-generated.
Alias	A human-readable nickname for this target.
Authentication mode	One of None, CHAP, or RADIUS.
CHAP name	If CHAP authentication is used, the CHAP username.
CHAP secret	If CHAP authentication is used, the CHAP secret.
Network interfaces	The interfaces whose target portals are used to export this target.

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

icon	description
	Target is online
	Target is offline

Clustering Considerations

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.

Initiator Configuration

iSCSI initiators have the following configurable properties.

Property	Description
Initiator IQN	The IQN for this initiator.
Alias	A human-readable nickname for this initiator.
Use CHAP	Enables or disables CHAP authentication
CHAP name	If CHAP authentication is used, the CHAP username.
CHAP secret	If CHAP authentication is used, the CHAP secret.

Planning Client Configuration

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
- 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.

Solaris iSCSI/iSER and MPxIO Considerations

MPxIO supports target port aggregation and availability in Solaris iSCSI configurations that configure multiple sessions per target (MS/T) on the iSCSI initiator.

- Use IPMP for aggregation and failover of two or more NICs.

- A basic configuration for an iSCSI host is a server with two NICs that are dedicated to iSCSI traffic. The NICs are configured by using IPMP. Additional NICs are provided for non-iSCSI traffic to optimize performance.
- Active multipathing can only be achieved by using the Solaris iSCSI MS/T feature, and the failover and redundancy of an IPMP configuration.
- *If one NIC fails in an IPMP configuration, IPMP handles the failover. The MPxIO driver does not notice the failure. In a non-IPMP configuration, the MPxIO driver fails and offlines the path.
- *If one target port fails in an IPMP configuration, the MPxIO driver notices the failure and provides the failover. In a non-IPMP configuration, the MPxIO driver notices the failure and provides the failover.
- For more information about using the Solaris iSCSI MS/T feature with IPMP and multipathing, see SunSolve Infodoc 207607, Understanding an iSCSI MS/T multi-path configuration.
- For information about configuring multiple sessions per target, see How to Enable Multiple iSCSI Sessions for a Target in the following document: http://docs.sun.com/app/docs/doc/817-5093/gcawf (`http://docs.sun.com/app/docs/doc/817-5093/gcawf`)
- For information about configuring IPMP, see Part VI, IPMP, in System Administration Guide: IP Services, in the following document: http://docs.sun.com/app/docs/doc/816-4554/ipmptm-1 (`http://docs.sun.com/app/docs/doc/816-4554/ipmptm-1`)

Troubleshooting

For tips on troubleshooting common iSCSI misconfiguration, see the iSCSI section.

Observing Performance

iSCSI performance can be observed via analytics, whereby one can breakdown operations or throughput by initiator, target, or LUN.

BUI

Creating an Analytics Worksheet

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.

CLI

Adding an iSCSI target with an auto-generated IQN

```
ahi:configuration san targets iscsi> create
ahi:configuration san targets iscsi target (uncommitted)> set alias="Target 0"
ahi:configuration san targets iscsi target (uncommitted)> set auth=none
ahi:configuration san targets iscsi target (uncommitted)> set interfaces=nge1
ahi:configuration san targets iscsi target (uncommitted)> commit
ahi:configuration san targets iscsi> list
TARGET     ALIAS
target-000 Target 0
           |
           +-> IQN
               iqn.1986-03.com.sun:02:daf0161f-9f5d-e01a-b5c5-e1efa9578416
```

Adding an iSCSI target with a specific IQN and RADIUS authentication

```
ahi:configuration san targets iscsi> create
ahi:configuration san targets iscsi target (uncommitted)> set alias="Target 1"
ahi:configuration san targets iscsi target (uncommitted)>
     set iqn=iqn.2001-02.com.acme:12345
ahi:configuration san targets iscsi target (uncommitted)> set auth=radius
ahi:configuration san targets iscsi target (uncommitted)> set interfaces=nge1
ahi:configuration san targets iscsi target (uncommitted)> commit
ahi:configuration san targets iscsi> 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 which uses CHAP authentication

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

```
     set chapsecret=123456789012abc
ahi:configuration san initiators iscsi initiator (uncommitted)> commit
ahi:configuration san initiators iscsi> list
NAME          ALIAS
initiator-000 Init 0
              |
              +-> INITIATOR
                  iqn.2001-02.com.acme:initiator12345
```

Adding an iSCSI target group

```
ahi:configuration san targets iscsi groups> create
ahi:configuration san targets iscsi group (uncommitted)> set name=tg0
ahi:configuration san targets iscsi group (uncommitted)>
    set targets=iqn.2001-02.com.acme:12345,
                iqn.1986-03.com.sun:02:daf0161f-9f5d-e01a-b5c5-e1efa9578416
ahi:configuration san targets iscsi group (uncommitted)> commit
ahi:configuration san targets iscsi groups> list
GROUP     NAME
group-000 tg0
          |
          +-> TARGETS
              iqn.2001-02.com.acme:12345
              iqn.1986-03.com.sun:02:daf0161f-9f5d-e01a-b5c5-e1efa9578416
```

Adding an iSCSI initiator group

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

SRP

Introduction

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).

Target configuration

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.

Property	Description
Target EUI	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.
Alias	A human-readable nickname for this target.

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

icon	description
	Target is online
	Target is offline

Clustering Considerations

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.

Initiator configuration

SRP initiators have the following configurable properties.

Property	Description
Initiator EUI	The EUI for this initiator.
Alias	A human-readable nickname for this initiator.

Observing Performance

SRP performance can be observed via analytics, whereby one can breakdown operations or throughput by initiator or target. {{Server}}/wiki/images/cfg_san_srp.png

Multipathing Considerations

The following sections provide a guide for setting up host clients.

Linux with OFED SRP Initiator

The following procedure describes how to setup OFED.

1. Download the Linux OFED packages from: http://www.openfabrics.org/download_linux.htm (`http://www.openfabrics.org/download_linux.htm`)

2. Run the `install.pl` script with `--all` option. Note that the the `all` option will install SRP and associated tools. If the command fails because of package dependency, review the results and install all the packages it asks for, including 'gcc', if listed.

3. After the install is complete, read all of the SRP Release notes, as follows:

- http://www.openfabrics.org/downloads/OFED/ofed-1.4/OFED-1.4-docs/srp_release_notes.txt (`http://www.openfabrics.org/downloads/OFED/ofed-1.4/OFED-1.4-docs/srp_release_notes.txt`)

4. The Release Notes recommend using the `-n` flag for all `srp_daemon` invocations.

- To execute SRP daemon as a daemon you may execute `run_srp_daemon`. Provide with it the same options used for running `srp_daemon`. This script is found under `/usr/local/ofed/sbin/` or `<prefix>/sbin/`. Be sure only one instance of `run_srp_daemon` runs per port.
- To execute SRP daemon as a daemon on all the ports, execute `srp_daemon.sh` (found under `/usr/local/ofed/sbin/` or `<prefix>/sbin/`). Note that `srp_daemon.sh` sends its log to `/var/log/srp_daemon.log`.

5. To configure this script to execute automatically when the InfiniBand driver starts, alter one of the values as follows:

- Change the value of SRP_DAEMON_ENABLE in `/etc/infiniband/openib.conf` to "yes".

OR

- Change the value of SRPHA_ENABLE in `/etc/infiniband/openib.conf` to "yes". Note that the latter option also enables SRP High Availability.

6. To use High-Availability - Automatic mode, perform the following:

- Edit `/etc/infiniband/openib.conf` and set SRPHA_ENABLE to "yes".
- Restart `multipathd` and OpenIB, or reboot the initiator system, as follows:

```
service restart multipathd

/etc/init.d/openibd stop
/etc/init.d/openibd start
```

7. To display general characteristics of each initiator-side IB HCA and Port:

```
ibstat
```

8. To display all the available SRP target IO Controllers on the network, use one of the following:

```
srp_daemon -a -o -v

ibsrpdm
```

9. To add SCSI devices corresponding to a particular target, connect to appropriate device directory:

```
cd /sys/class/infiniband_srp/srp-mthca0-1
```

10. Enumerate the remote IO Controllers in a format that `add_target` will expect using one of the following, use `-n` below if you want to set the initiator-extension automatically:

```
srp_daemon -o -c
```

OR

```
ibsrpdm -c
```

11. Echo the appropriate line of output onto the system file add_target:

```
echo id_ext=0003ba0001002eac,ioc_guid=0003ba0001002eac,\
dgid=fe8000000000000003ba0001002ead,\
pkey=ffff,service_id=0003ba0001002eac > add_target
```

12. Use contents of `/var/log/messages` to determine which scsi device corresponds to each added target

- On some Linux, lsscsi should show the newly created device. Using the -H and -v options gives more device information, alternatively run:

```
cat /proc/scsi/scsi
```

13. Now you can do something with the device, such as:

```
mkfs /dev/sdd
```

For more information see Configuring Linux Initiators

OFED 1.5 Issues

- Degradation of performance noticed when stand-by path is activated.

This can happen when the I/O path to the active target is interrupted via a link failure or cluster takeover.

Workaround: To resume performance, I/O should be stopped on the initiator and a new session established to the target. From that point, I/O may be started and continue with the original level of throughput.

- Linux SRP initiator may keep session open after a cluster takeover and multi-path failover.

I/Os will continue but the srp_daemon will report that it has not connected to the takeover target. This problem can be seen by running:

```
srp_daemon -o -c -n -i <ib-device> -p <port-num>
```

The `srp_daemon` will report those targets to which it has not already connected. The expected behavior is for the `srp_daemon` to show the failed controller.

Workaround: Restart the `srp_daemon`.

- Session may hang during cluster takeover.

The client message log will report SRP failures and I/O errors in `/var/log/messages`:

```
Jan 27 11:57:03 ib-client-2 kernel:  host11: SRP abort called
Jan 27 11:57:37 ib-client-2 kernel:  host11: ib_srp: failed send status 12
Jan 27 11:57:37 ib-client-2 kernel: ib_srp:  host11: add qp_in_err timer
Jan 27 11:57:37 ib-client-2 kernel:  host11: ib_srp: failed send status 5
Jan 27 11:57:38 ib-client-2 kernel:  host11: SRP abort called
Jan 27 11:57:38 ib-client-2 kernel:  host11: SRP reset_device called
Jan 27 11:57:38 ib-client-2 kernel:  host11: ib_srp: SRP reset_host called state 0 qp_err 1
Jan 27 11:57:58 ib-client-2 kernel:  host11: SRP abort called
Jan 27 11:57:58 ib-client-2 kernel:  host11: SRP reset_device called
Jan 27 11:57:58 ib-client-2 kernel:  host11: ib_srp: SRP reset_host called state 0 qp_err 1
Jan 27 11:58:02 ib-client-2 kernel:  host11: ib_srp: srp_qp_in_err_timer called
Jan 27 11:58:02 ib-client-2 kernel:  host11: ib_srp: srp_qp_in_err_timer flushed reset - done
Jan 27 11:58:02 ib-client-2 kernel:  host11: ib_srp: Got failed path rec status -22
Jan 27 11:58:02 ib-client-2 kernel:  host11: ib_srp: Path record query failed
Jan 27 11:58:02 ib-client-2 kernel:  host11: ib_srp:
reconnect failed (-22), removing target port.
Jan 27 11:58:08 ib-client-2 kernel: scsi 11:0:0:0: scsi:
Device offlined - not ready after error recovery
Jan 27 11:58:08 ib-client-2 multipathd: sdc: tur checker reports path is down
Jan 27 11:58:08 ib-client-2 multipathd: checker failed path 8:32 in map mpath148
Jan 27 11:58:08 ib-client-2 multipathd: mpath148: Entering recovery mode: max_retries=200
Jan 27 11:58:08 ib-client-2 multipathd: mpath148: remaining active paths: 0
Jan 27 11:58:08 ib-client-2 multipathd: sdc: remove path (uevent)
Jan 27 11:58:08 ib-client-2 multipathd: mpath148: map in use
Jan 27 11:58:08 ib-client-2 multipathd: mpath148: can't flush
Jan 27 11:58:08 ib-client-2 multipathd: mpath148: Entering recovery mode: max_retries=200
Jan 27 11:58:08 ib-client-2 multipathd: dm-2: add map (uevent)
Jan 27 11:58:08 ib-client-2 multipathd: dm-2: devmap already registered
Jan 27 11:58:08 ib-client-2 kernel: scsi 11:0:0:0: scsi:
Device offlined - not ready after error recovery
Jan 27 11:58:08 ib-client-2 last message repeated 49 times
Jan 27 11:58:08 ib-client-2 kernel: scsi 11:0:0:0: rejecting I/O to dead device
Jan 27 11:58:08 ib-client-2 kernel: device-mapper: multipath: Failing path 8:32.
Jan 27 11:58:08 ib-client-2 kernel: scsi 11:0:0:0: rejecting I/O to dead device
```

Workaround The client must be rebooted to recover the SRP service.

- Device mapper state may become out of date preventing devices being added to the map table.

When this problem occurs, the `/var/log/messages` log will show:

```
device-mapper: table: 253:2: multipath: error getting device
device-mapper: ioctl: error adding target to table
```

The `multipath` command queries will report the correct state:

```
ib-client-1:~ # multipath -d
reload: maguro2LUN (3600144f08068363800004b6075db0001)
n/a SUN,Sun Storage 7310
[size=40G][features=0][hwhandler=0][n/a]
\_ round-robin 0 [prio=50][undef]
 \_ 18:0:0:1 sde 8:64  [undef][ready]
\_ round-robin 0 [prio=1][undef]
 \_ 17:0:0:1 sdc 8:32  [failed][ghost]
```

Workaround: The client must be rebooted to clear the stale device mapper state.

- SRP initiator may go into an infinite retry target loop.

Upon cluster takeover, a multipath device may not swtich to the standby path as expected. The problem is with the SRP initiator. The initiator is in an infinite loop write, fail, abort, reset device, reset target operations. When this problem happens, the following messages will be logged in the `/var/log/messages` log:

```
Jan 26 17:42:12 mysystem kernel: sd 13:0:0:0: [sdd]
Device not ready: Sense Key : Not Ready [current]
Jan 26 17:42:12 mysystem kernel: sd 13:0:0:0: [sdd]
Device not ready: Add. Sense: Logical unit not accessible, target port in standby state
Jan 26 17:42:12 mysystem kernel: end_request: I/O error, dev sdd, sector 512248
Jan 26 17:42:12 mysystem kernel: scsi host13: SRP abort called
Jan 26 17:42:12 mysystem kernel: scsi host13: SRP reset_device called
Jan 26 17:42:12 mysystem kernel: scsi host13: ib_srp: SRP reset_host called state 0 qp_err 0
Jan 26 17:42:21 mysystem multipathd: 8:48: mark as failed
```

Workaround: Remove the device and re-scan.

VMWare 4.0

The VMware Native MultiPath Plugin (nmp) has two components that can be changed on a device by device, path by path, or array basis.

Path Selection Plugin (psp)

Controls which physical path is used for I/O

```
# esxcli nmp psp list
Name           Description
VMW_PSP_MRU    Most Recently Used Path Selection
VMW_PSP_RR     Round Robin Path Selection
VMW_PSP_FIXED  Fixed Path Selection
```

Storage Array Type Plugin (satp)

Controls how failover works

The SATP has to be configured to recognize the array vendor or model string in order to change the basic failover mode from a default Active/Active type array to ALUA.

By default the Sun Storage 7000 cluster was coming up as a Active/Active array only.

To manually change this you need to use the ESX CLI to add a rule to have the ALUA plugin claim the 7000 luns.

```
# esxcli nmp satp addrule -s VMW_SATP_ALUA \
   -e "Sun Storage 7000" -V "SUN" -M "Sun Storage 7410" -c "tpgs_on"

options are:

-s VMW_SATP_ALUA - for the ALUA SATP
-e description of the rule
-V Vendor
-M Model
-c claim option for Target Portal Group (7000 seems to support implicit)
```

If no luns have been scanned/discovered, you can simply rescan the adapter to find new luns, they should be claimed by the ALUA plugin. If luns are already present, reboot the ESX host.

After the reboot, we should see the luns being listed under the VMW_SATP_ALUE array type.

```
# esxcli nmp device list

naa.600144f096bb823800004b707f2d0001
Device Display Name: Local SUN Disk (naa.600144f096bb823800004b707f2d0001)
Storage Array Type: VMW_SATP_ALUA
Storage Array Type Device Config:
   {implicit_support=on;explicit_support=off;explicit_allow=on;
    alua_followover=on; {TPG_id=0,TPG_state=AO}{TPG_id=1,TPG_state=STBY}}
Path Selection Policy: VMW_PSP_MRU
Path Selection Policy Device Config: Current Path=vmhba_mlx4_1.1.1:C0:T1:L0
Working Paths: vmhba_mlx4_1.1.1:C0:T1:L0
```

Relevant lun path lists should show a Active and a Standby path

```
# esxcli nmp path list

gsan.80fe53553e0100282100-gsan.80fe8f583e0100282100
-naa.600144f096bb823800004b707f2d0001
Runtime Name: vmhba_mlx4_1.1.1:C0:T2:L0
Device: naa.600144f096bb823800004b707f2d0001
Device Display Name: Local SUN Disk (naa.600144f096bb823800004b707f2d0001)
Group State: standby
Storage Array Type Path Config:
{TPG_id=1,TPG_state=STBY,RTP_id=256,RTP_health=UP}
Path Selection Policy Path Config: {non-current path}

gsan.80fe53553e0100282100-gsan.80fe73583e0100282100
-naa.600144f096bb823800004b707f2d0001
Runtime Name: vmhba_mlx4_1.1.1:C0:T1:L0
Device: naa.600144f096bb823800004b707f2d0001
Device Display Name: Local SUN Disk (naa.600144f096bb823800004b707f2d0001)
Group State: active
Storage Array Type Path Config:
{TPG_id=0,TPG_state=AO,RTP_id=2,RTP_health=UP}
Path Selection Policy Path Config: {current path}
```

VMWare ESX 4.0 Issues

- Standby and active paths may not be found

The `esxcl nmp path list` command should report an active and a standby path one each for the SRP targets in a cluster configuration.

```
[root@ib-client-5 vmware]# esxcli nmp path list
gsan.80fe53553e0100282100-gsan.80fe8f583e0100282100-
naa.600144f096bb823800004b707f2d0001
   Runtime Name: vmhba_mlx4_1.1.1:C0:T2:L0
   Device: naa.600144f096bb823800004b707f2d0001
   Device Display Name: Local SUN Disk
(naa.600144f096bb823800004b707f2d0001)
   Group State: standby
   Storage Array Type Path Config:
{TPG_id=1,TPG_state=STBY,RTP_id=256,RTP_health=UP}
   Path Selection Policy Path Config: {non-current path}

gsan.80fe53553e0100282100-gsan.80fe73583e0100282100-
   naa.600144f096bb823800004b707f2d0001
   Runtime Name: vmhba_mlx4_1.1.1:C0:T1:L0
   Device: naa.600144f096bb823800004b707f2d0001
   Device Display Name: Local SUN Disk
(naa.600144f096bb823800004b707f2d0001)
   Group State: active
   Storage Array Type Path Config:
{TPG_id=0,TPG_state=AO,RTP_id=2,RTP_health=UP}
   Path Selection Policy Path Config: {current path}
```

When this problem occurs, the active or standby path may not be shown in the output of `esxcli nmp path list`.

Workaround: None

- VMWare VM Linux guest may hang during cluster takeover

When this problem happens, the Linux guest system log will report in its `/var/log/messages` log:

```
Feb 10 16:10:00 ib-client-5 vmkernel: 1:21:41:36.385 cpu3:4421)<3>ib_srp:
Send tsk_mgmt target[vmhba_mlx4_1.1.1:2] out of TX_IU head 769313 tail 769313 lim 0
```

Workaround: Reboot guest VM

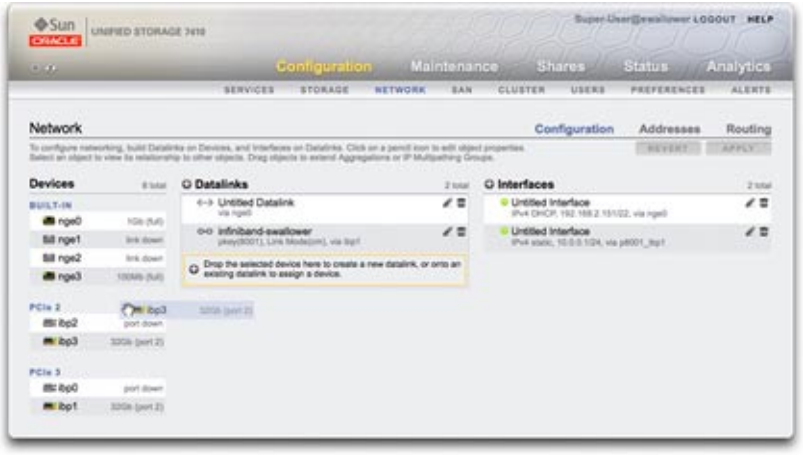
BUI

This section describes instructions for using the BUI to configure iSER and SRP targets and initiators.

iSER Target Configuration

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

1. To configure ibd interfaces, select the ibd interface (or ipmp), and drag it to the Datalinks list to create the datalink on the Configuration > Network screen. Then drag the Datalink to the Interfaces list to create a new interface.



2. To create an iSER target, got to the Configuration > SAN screen. Click the iSCSI Targets link and then click the add icon to add a new iSER target with an alias.

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

4. To create an initiator, click the Initiator link and then click the iSCSI initiators link. Click the add icon to add a new initiator. Enter the Initiator IQN and an alias and click OK.

While creating an initiator group is optional, if you don't create a group, the LUN associated with the target will be available to all initiators. To create a group, drag the initiator to the iSCSI Initiator Groups list.

5. To create a LUN, go to the Shares screen and click the LUN link. Then 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.

6. Two client initiators are supported: RedHat 4 and SUSE 11, use any method to discover iSER LUNs on the client.

SRP Target Configuration

This procedure describes the steps for configuring SRP targets.

1. Connect HCA ports to IB interfaces.

The targets are automatically discovered by the appliance.

2. To create the target group, go to the Configuration > SAN screen.

3. Click the Target link and then click SRP targets

The SRP targets page appears.

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

5. (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.

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

7. 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.

The following SRP initiators have been tested and are known to work:

- VMWare ESX
- RedHat 5.4
- SUSE11

CLI

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

```
swallower:configuration san targets srp groups> create
swallower:configuration san targets srp group (uncommitted)> set name=targetSRPgroup
                          name = targetSRPgroup (uncommitted)
swallower:configuration san targets srp group (uncommitted)>
set targets=eui.0002C903000489A4
                       targets = eui.0002C903000489A4 (uncommitted)
swallower:configuration san targets srp group (uncommitted)> commit
swallower:configuration san targets srp groups> list
GROUP     NAME
group-000 targetSRPgroup
```

```
        |
        +-> TARGETS
            eui.0002C903000489A4
```

The following example demonstrates 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
```

Users

Introduction

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 - all their account information is saved on the appliance.
- Directory users - this uses existing NIS or LDAP accounts, and saves supplemental authorization settings on the appliance. This allows existing NIS or LDAP users to be granted privileges to login and administer the appliance.

Users are granted privileges by assigning them custom *roles*.

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 Audit log.

By default, a role called "Basic administration" exists, which contains very basic authorizations.

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.

Available scopes are as follows, with a single example authorization and an example filter (if available) for each scope:

Scope	Example Authorization	Example Filter
Active Directory	Join an Active Directory domain	Domain name
Alerts	Configure alert filters and thresholds	.
Analytics	Read a statistic with this drilldown present	Drilldowns
Clustering	Failback resources to a cluster peer	.
Hardware	Online and offline disks	.
Networking	Configure networking devices, datalinks, and interfaces	.
Projects and shares	Change general properties of projects and shares	Pool, project, share
Roles	Configure authorizations for a role	Role name
Services	Restart a service	Service name
Shares property schema	Modify property schema	.
System	Reboot the appliance	Appliance name
Update	Update system software	.
Users	Change a password	Username
Worksheet	Modify worksheet	Worksheet name

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.

Properties

The following properties may be set when managing users and roles.

Users

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

Property	Description
Type	Directory (access credentials from NIS or LDAP), or Local (save user on this appliance)
Username	Unique name for user
Full Name	User description
Password/Confirm	For Local users, type the initial password in both of these fields
Require session annotation	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 Audit log.
Kiosk user	If enabled, the user will only be able to view the screen in the "Kiosk screen" setting. This may be used for restrict a user to only see the dashboard, for example. A kiosk user will not be able to access the appliance via the CLI.
Kiosk screen	Screen that this kiosk user is restricted to, if "Kiosk user" is enabled
Roles	The roles possessed by this user
Exceptions	These authorizations are excluded from those normally available due to the selected roles

Roles

These properties may be set when managing roles:

Property	Description
Name	Name of the role as it will be shown in lists
Description	Verbose description of role if desired
Authorizations	Authorizations for this role

BUI

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:

icon	description
	Add new user/role. This will display a new dialog where the required properties may be entered.
	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 "Show All" to return to the full listings.
	Clone user/role. Add a new user/role starting with fields based on the values from this entry
	Edit user/role
	Remove user/role/authorization

Refer to the Tasks for required steps to add users, roles and authorizations.

CLI

The actions possible in the BUI are also available in the CLI. Type `help` as you navigate through user, role, and authorization administration to list the available commands.

To demonstrate the CLI user and roles interface, the following example adds the NIS user "brendan" to the system, and grants the authorization to restart the HTTP service. This includes creating a role for this authorization.

We will start by creating the role, which we will call "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
```

Now that we have created the webadmin role, we will add 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:

```
caji:configuration roles> select webadmin
caji:configuration roles webadmin> authorizations
caji:configuration roles webadmin authorizations> create
caji:configuration roles webadmin auth (uncommitted)> set scope=tab
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:nge0   idmap           ndmp            scrk
dns             interface:nge0  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
caji:configuration roles webadmin authorizations> list
NAME       OBJECT                              PERMISSIONS
auth-000   svc.http                            restart
```

Now that the role has been created, we can enter the users section to create our user "brendan" and assign the role "webadmin":

```
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
```

The user brendan should now be able to login using their NIS password, and restart the HTTP service on the appliance.

Tasks

The following are example tasks for user and role administration. If you wish to use the CLI, it can help to practice these tasks in the BUI first - which is more intuitive and will help convey concepts.

BUI

▼ Adding an administrator

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

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

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

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.

CLI

▼ Adding an administrator

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 and `set` them, 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. Roles and authorization exceptions may now be added, as well as user preferences.

▼ Adding a role

1. Go to `configuration roles`.
2. Type `role` followed by the role name you wish to create.
3. Set the description, then `commit` the role.
4. Add authorizations to the role (see separate task).

▼ Adding authorizations to a role

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. `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.

▼ Deleting authorizations from a role

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.

Generic

▼ Adding a user who can only view the dashboard

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.

Preferences

Introduction

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

Property	Description
Initial login screen	First page the BUI will load after a successful login. By default this is the Status Dashboard.
Locality	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, Galician, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish and Swedish.
Session timeout	Time after navigating away from the BUI that the browser will automatically logout the session
Current session annotation	Annotation text added to audit logs
Advanced analytics statistics	This will make available additional statistics in Analytics
SSH Public Keys	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.

BUI

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

CLI

Preferences may be set in the CLI under `configuration users`. 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
```

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
                   login_screen = status/dashboard
                session_timeout = 15
             session_annotation =
             advanced_analytics = false

Children:
                              keys => Manage SSH public keys

twofish:configuration preferences> set session_annotation="Editing my user preferences"
             session_annotation = Editing my user preferences (uncommitted)
twofish:configuration preferences> commit
```

SSH Public Keys

These may be needed when automating the execution of CLI scripts from another host. The following shows the addition of an SSH key from the CLI:

```
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
```

The key text is just the key text itself (usually hundreds of characters), without spaces.

Alerts

Introduction

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.

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

Alerts are grouped into the following categories:

Category	Description
Cluster	Cluster events, including link failures and peer errors
Custom	Events generated from the custom alert configuration
Hardware Events	Appliance boot and hardware configuration changes
Hardware Faults	Any hardware fault
NDMP operations	Backup and restore, start and finished events. This group is available as "NDMP: backup only" and "NDMP: restore only", for just backup or restore events
Network	Network port, datalink, and IP interface events and failures
Phone home	Support bundle upload events
Remote replication	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
Service failures	Software Service failure events
Thresholds	Custom alerts based on Analytics statistics
ZFS pool	Storage pool events, including scrub and hot space activation

Actions

The following actions are supported.

Send Email

An email containing the alert details can be sent. The configuration requires an email address and email subject line. The following is a sample email sent based on a threshold alert:

```
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: 0809QAU005, HOSTNAME: caji
SOURCE: svc:/appliance/kit/akd:default, REV: 1.0
EVENT-ID: 15a53214-c4e7-eae4-dae6-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-dae6-a652a51ea29b
```

Details on how the appliance sends mail can be configured on the SMTP service screen.

Send 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 is an example 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.
48.99.49.45.52.99.49.57.45.101.57.99.98.45.97.99.50.55.102.55.49.50.
54.98.55.57 = STRING: "cpu.utilization threshold of 90 is violated."
```

Send Syslog Message

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

Resume/Suspend Dataset

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.

Resume/Suspend Worksheet

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.

Threshold Alerts

These are alerts based on the statistics from Analytics. The following are properties when creating threshold alerts:

Property	Description
Threshold	The threshold statistic is from Analytics, and is self descriptive (eg, "Protocol: NFSv4 operations per second")
exceeds/falls below	defines how the threshold value is compared to the current statistic
Timing: for at least	Duration which the current statistic value must exceed/fall below the threshold
only between/only during	These properties may be set so that the threshold is only sent during certain times of day - such as business hours

Property	Description
Repost alert every ... this condition persists.	If enabled, this will re-execute the alert action (such as sending email) every set interval while the threshold breech exists
Also post alert when this condition clears for at least ...	Send a followup alert if the threshold breech clears for at least the set interval

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

BUI

At the top of the Configuration->Alerts page are tabs for "Alert Actions" and "Threshold Alerts". See the Tasks for step by step instructions for configuring these in the BUI.

CLI

Alerts can also be configured from the CLI. Enter the `configuration alerts` and type `help`.

Tasks

BUI

▼ Adding an alert action

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

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

2 Pick the statistic to monitor. You can use Analytics 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.

Workflows

Introduction

A *workflow* is a 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 actions or at a designated time. As such, workflows allow for the appliance to *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.

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:

Required member	Type	Description
`name`	String	Name of the workflow
`description`	String	Description of workflow
`execute`	Function	Function that executes the workflow

Here is the canonically trivial workflow:

```
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!"

Workflow execution context

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, 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:

```
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] + '%');
       }
};
```

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:

Required Member	Type	Description
`label`	String	Label to adorn input of workflow parameter
`type`	String	Type of workflow parameter

The `type` member must be set to one of these types:

Type name	Description
`Boolean`	A boolean value
`ChooseOne`	One of a number of specified values
`EmailAddress`	An e-mail address
`File`	A file to be transferred to the appliance
`Host`	A valid host, as either a name or dotted decimal
`HostName`	A valid hostname
`HostPort`	A valid, available port
`Integer`	An integer
`NetAddress`	A network address
`NodeName`	A name of a network node
`NonNegativeInteger`	An integer that is greater than or equal to zero
`Number`	Any number -- including floating point
`Password`	A password
`Permissions`	POSIX permissions
`Port`	A port number
`Size`	A size
`String`	A string
`StringList`	A list of strings

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):

```
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 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:

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`

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:

```
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 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:

Optional Member	Type	Description
`optional`	Boolean	If set to `true`, 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.

If a parameter is optional and is unset, its member in the parameters object passed to the `execute` function will be set to `undefined`.

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. This example could be modified to catch the offending error, and create the project in the case that it doesn't exist:

```
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 + '"');
    }
};
```

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. To extend our example to give a crisp error if the user attempts to create an extant share:

```
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 + '"');
    }
};
```

Execution auditing

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.

Execution reporting

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:

Required Member	Type	Description
`step`	String	String that denotes the name of the execution step
`execute`	Function	Function that executes the step of the workflow

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. As an example, here is a workflow that creates a new project, share, and audit record over three steps:

```
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); }
};
```

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:

Optional Member	Type	Description
`required`	String	The minimum version of the appliance software required to run this workflow, including the minimum year, month, day, build and branch.
`version`	String	Version of this workflow, in dotted decimal (major.minor.micro) form.

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 build 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 view in the BUI. Here's an example of using the `required` field:

```
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".

Workflows as alert actions

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`.

Alert action execution context

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:

Required Member	Type	Description
`class`	String	The class of the alert.
`code`	String	The code of the alert.
`items`	Object	An object describing the alert.
`timestamp`	Date	Time of alert.

The `items` member of the parameters object has the following members:

Required Member	Type	Description
`url`	String	The URL of the web page describing the alert
`action`	String	The action that should be taken by the user in response to the alert.
`impact`	String	The impact of the event that precipitated the alert.
`description`	String	A human-readable string describing the alert.
`severity`	String	The severity of the event that precipitated the alert.

Auditing alert actions

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:

```
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 aberrent 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');
        }
};
```

Example: device type selection

Here is an example workflow that creates a worksheet based on a specified drive type:

```
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 drilldowns=' + 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); }
};
```

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.

CLI

Workflows are manipulated in the `maintenance workflows` section of the CLI.

Downloading workflows

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)
```

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
```

Viewing workflows

To list workflows, use the `list` command from the `maintenance workflows` context:

```
dory:maintenance workflows> list
WORKFLOW     NAME                OWNER      SETID ORIGIN
workflow-000 Hello world         root       false <local>
```

To select a workflow, use the `select` command:

```
dory:maintenance workflows> select workflow-000
dory:maintenance workflow-000>
```

To get a workflow's properties, use the `get` command from within the context of the selected workflow:

```
dory:maintenance workflow-000> get
                        name = Hello world
                 description = Bids a greeting to the world
                       owner = root
                      origin = <local>
                       setid = false
                       alert = false
               scheduled = false
```

Executing workflows

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!
```

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)
```

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"
```

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
```

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
```

Cluster

Clustering

The Sun ZFS Storage 7000 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 Sun ZFS Storage 7000 series appliances. Because the clustering feature relies on shared access to storage resources, it is available only on the Sun ZFS Storage 7310, 7320, 7410, 7420 and 7720. You will be unable to configure clustering on other appliance models, or if the two heads are not of the same model.

This section is presented in several segments, beginning with background material helpful in the planning process. Understanding this material is critical to performing the configuration and maintenance tasks described in later segments and more generally to a successful unified storage deployment experience.

Features and Benefits

It is important to understand the scope of the Sun ZFS Storage 7000 series 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 âheadsâ. 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 âactive-activeâ 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 âactive-passiveâ to refer to which a single storage pool is assigned to the head designated as âactiveâ along with its associated network interfaces. Both topologies are supported by the Sun ZFS Storage 7000 System. 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 âpeerâ) 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 Sun ZFS Storage 7000 series 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 Sun ZFS Storage 7000 product family and the data protocols if offers support numerous other features and strategies that can improve availability:

- Remote 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, and
- Redundant storage devices (RAID).

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

Drawbacks

When deciding between a clustered and standalone Sun ZFS Storage 7000 series 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 Sun ZFS Storage 7000 series), 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 a Sun ZFS Storage 7000 series 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.

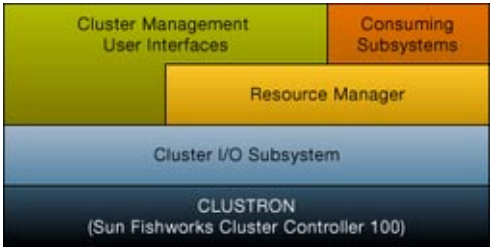
Terminology

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_STRIPPED state to AKCS_OWNER, in which all resources are imported

Subsystem Design

The clustering subsystem incorporated into the Sun ZFS Storage 7000 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.



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 CLUSTRON 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 Sun ZFS Storage 7000 series 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.

Clustered heads never communicate using external service or administration network interfaces, and the interconnects form a secure private network. 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 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 Sun ZFS Storage System 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.

Resource Management Concepts

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 âPrimaryAâ interface will always be provided by the head that has imported pool-0, while the addresses associated with âPrimaryBâ 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 discussion of specific use cases in that section below.

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.

Resource	icon	Omnipresent	Taken over on failure
SINGLETON		No	Yes
REPLICA	None	Yes	N/A
PRIVATE		No	No
SYMBIOTE	None	Same as parent type	Same as parent type

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.

Takeover and Failback

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

State	icon	CLI/BUI Expression	Description
UNCONFIGURED		Clustering is not configured	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.
OWNER		Active (takeover completed)	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 take-over). It remains in this state until an administrator manually executes a fail-back operation.

State	icon	CLI/BUI Expression	Description
STRIPPED		Ready (waiting for failback)	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.
CLUSTERED		Active	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.
-		Rejoining cluster ...	The appliance has recently rebooted, or the appliance management software is restarting after an internal failure. Resource state is being resynchronized.
-		Unknown (disconnected or restarting)	The peer appliance is powered off or rebooting, all its cluster interconnect links are down, or clustering has not yet been configured.

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 not necessary â indeed, it is redundant â 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:

1. 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.
2. Make all other changes on either head, but not both. Site policy should specify which head is to be considered the âmasterâ 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 âamnesiaâ, 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 Sun ZFS Storage 7000 series, 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 a Sun ZFS Storage 7000 series system 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 the Storage Configuration documentation. 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.

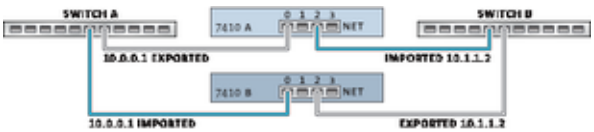
Variable	Single Node ownership	Multiple pools owned by different heads
Total throughput (nominal operation)	Up to 50% of total CPU resources, 50% of DRAM, and 50% of total network connectivity can be used to provide service at any one time. This is straightforward: only a single head is ever servicing client requests, so the other is idle.	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).
Total throughput (failed over)	No change in throughput relative to nominal operation.	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.
I/O latency (failed over)	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.	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.
Storage flexibility	All available physical storage can be used by shares and LUNs.	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 pool fills up while the other has free space, some storage may be wasted.

Variable	Single Node ownership	Multiple pools owned by different heads
Network connectivity	All network devices in each head can be used while that head is providing service. In the 7410C, up to three expansion slots plus 4 built-in network devices can be used concurrently to provide connectivity to the single pool.	Only half of all network devices in each head can be used while that head is providing service. Therefore each pool can be connected to only half as many physically disjoint networks.

A second important consideration for storage is the use of pool configurations with no single point of failure (âNSPFâ). Since the use of clustering implies that the application places a very high premium on availability, there is seldom a good reason to configure storage pools in a way that allows the failure of a single JBOD to cause loss of availability. The downside to this approach is that NSPF configurations require a greater number of JBODs than do configurations with a single point of failure; when the required capacity is very small, installation of enough JBODs to provide for NSPF at the desired RAID level may not be economical.

Clustering Considerations for Networking

Network device, datalink, and interface failures do not cause the clustering subsystem to consider a head to have failed. To protect against network failures â whether inside or outside the appliance â IPMP and/or LACP should be used instead. These network configuration options, along with a broader network-wide plan for redundancy, are orthogonal to clustering and are additional components of a comprehensive approach to availability improvement.



Network interfaces may be configured as either singleton or private resources, provided they have static IP configuration (interfaces configured to use DHCP can only be private; the use of DHCP in clusters is discouraged). When configured as a singleton resource, all of the datalinks and devices used to construct an interface may be active on only one head at any given time. Likewise, corresponding devices on each head must be attached to the same networks in order for service to be provided correctly in the failed-over state. A concrete example of this is shown in Illustration 5. When constructing network interfaces from devices and datalinks, it is essential to proper cluster operation that each singleton interface have a device with the same identifier and capabilities available on both heads. Since device identifiers depend on the device

type and the order in which it is first detected by the appliance, any two clustered heads MUST have identical hardware installed. Furthermore, 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 Sun reseller or service representative can assist in planning hardware upgrades that will meet these requirements.

A route is always bound explicitly to a single network interface. Routes are represented within the resource manager as replicas, but can become active only when the interfaces they are bound to are operational. Therefore, a route bound to an interface that is currently in standby mode (exported) will have no effect until that interface is activated during the process of takeover. This becomes important when two pools are configured and must be made available to a common subnet. In this case, if that subnet is home to a router that should be used by the appliances to reach one or more other networks, then 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.

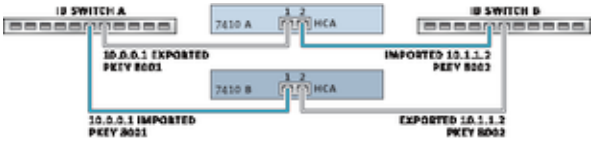
- Example: Interface e1000g3 is assigned to 'alice' and e1000g4 is assigned to 'bob'. Each interface has an address in the 172.16.27.0/24 network and will be used to provide service to clients in the 172.16.64.0/22 network, reachable via 172.16.27.1. Two routes should be created to 172.16.64.0/22 via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

It is often advantageous to assign each clustered head an IP address â most likely on a dedicated management network â to be used only for administration, and to designate as a private resource the interface on which this address is configured. This ensures that it will be possible to reach any functioning head from the management network, even if it is currently in the AKCS_STRIPPED state and awaiting failback. This is especially important if services such as LDAP and Active Directory are in use that require access to other network resources even when the head is not itself providing service. If this is not practical, it is especially important that the service processor 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 will be impossible to manage or monitor a newly-booted head until failback has completed. Conversely, the need may also arise to monitor or manage whichever head is currently providing service (or service for a particular storage pool). This is most likely to be useful when it is necessary to modify some aspect of the storage itself; e.g., to modify a share property or create a new LUN. This can be achieved either by using one of the service interfaces to perform administrative tasks or by allocating a separate singleton interface to be used only for the purpose of managing the pool to which it is matched. In either case, the interface should be assigned to the same head as the pool it will be used to manage.

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

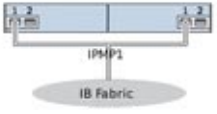
appliance. The network topology should include IPMP to protect against network failures at the link level with a broader plan for redundancy for HCAs, switches and subnet managers.



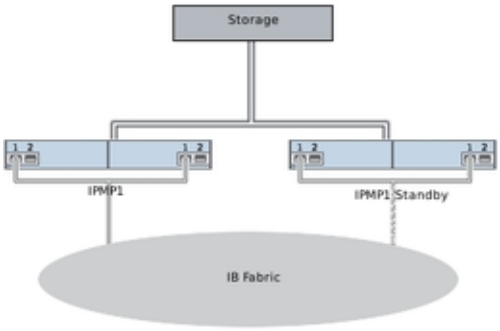
To ensure proper cluster configuration, each head must be populated with identical HCAs in identical slots. Furthermore, each corresponding HCA port must be configured into the same partition (pkey) on the subnet manager with identical membership privileges and attached to the same network. To reduce 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 Sun reseller or service representative can assist in planning hardware upgrades that will meet these requirements.

Redundant Path Scenarios

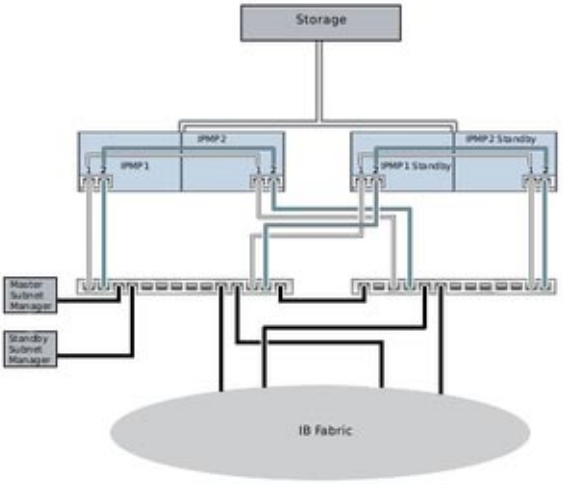
The following illustration shows HCA port link redundancy on the 7110. Redundancy at the port level is such that if any single IB port fails, none of the other ports have interrupted service.



The following illustration shows cluster configuration on the 7410 for host redundancy.



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.



Preventing "Split-Brain" Conditions

A common failure mode in clustered systems is known as âsplit-brainâ; 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 Sun ZFS Storage 7000 series appliances, this would occur if the cluster I/O links fail. In addition to the built-in triple-link

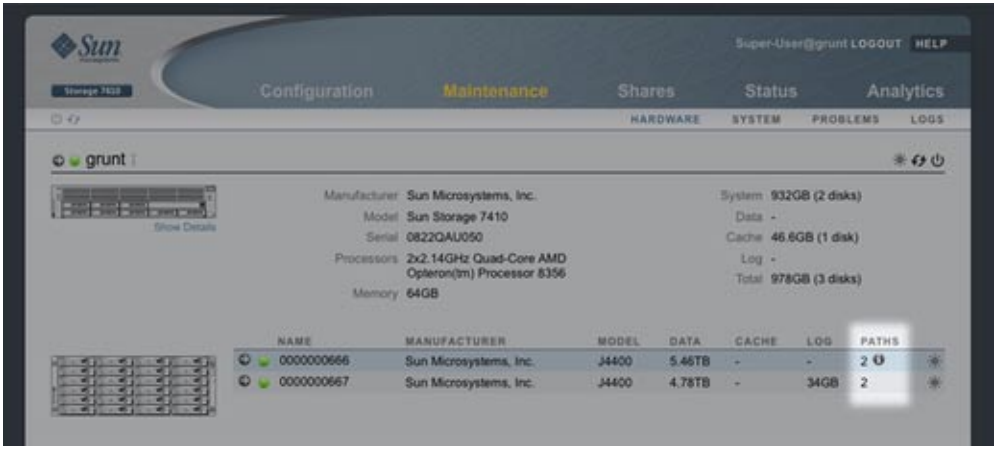
redundancy (only a single link is required to avoid triggering takeover), the appliance software will also perform an arbitration procedure to determine which head should continue with takeover.

A number of arbitration mechanisms are employed by similar products; typically they entail the use of âquorum disksâ (using SCSI reservations) or âquorum serversâ. To support the use of ATA disks without the need for additional hardware, the Sun ZFS Storage 7000 series 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 JBOD 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 Sun ZFS Storage 7000 series appliances 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 JBOD (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 JBOD 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.



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 Sun ZFS Storage 7000 family of appliances (see illustration); reducing it will reduce the apparent impact of takeover and/or failback.

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 JBOD, provided the disk bays

in that half-JBOD 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 â on the order of tens of milliseconds â 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 Sun ZFS Storage 7000 series clustered appliance in a production environment.

Setup Procedure

When setting up a cluster from two new appliances, perform the following steps:

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 JBOD(s) as shown in the JBOD Cabling diagrams in the setup poster that came with your Sun ZFS Storage system.

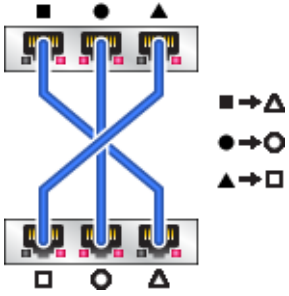
4. Power on both appliances - but do not begin configuration. Select only one of the two appliances from which you will perform configuration; the choice is arbitrary. This will be referred to as the primary appliance for configuration purposes. Connect to and access the serial console of that appliance, 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. Note: 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:
9. # 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.
10. # 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.
11. # 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.
12. 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. Note: The smallest unit of storage that may be assigned to a pool is half a JBOD. Therefore, if you have only a single JBOD and wish to create two pools, you must use the popup menu to select Half of your JBOD for each pool. Additionally, it is not possible to create two pools if you have attached only a single half-populated JBOD. If you choose to create two pools, there is no requirement that they be the same size; any subdivision of available storage is permitted.

13. 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.
14. 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.

Node Cabling

Clustered head nodes must be connected together using the cluster interconnect controller. This device is installed in slot PCIe0 in the Sun Storage 7310 and ZFS Storage 7320. The cluster controller is installed in slot PCIe5 in the Sun Storage 7410 and slot PCIeC in the ZFS Storage 7420/7720.



The controller provides 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.

JBOD Cabling

You will need to attach your JBODs to both appliances before beginning cluster configuration. See Installation: Cabling Diagrams or follow the Quick Setup poster that shipped with your system.

BUI

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 these 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 interface contains these buttons:

Button	Description
Setup	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.
Unconfig	Upgrade a node to standalone operation by unconfiguring the cluster. See below for a detailed description of this task.
Apply	If resource modifications are pending (rows highlighted in yellow), commit those changes to the cluster.
Revert	If resource modifications are pending (rows highlighted in yellow), revert those changes and show the current cluster configuration.
Failback	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).
Takeover	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

Unconfiguring Clustering

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 perform cluster unconfiguration:

1. You no longer wish to use clustering; instead, you wish to configure two independent storage appliances.
2. You are replacing a failed storage controller with new hardware or a storage controller with factory-fresh appliance software (typically this replacement will be performed by your service provider).

The steps for unconfiguring a cluster are as follows:

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. **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.

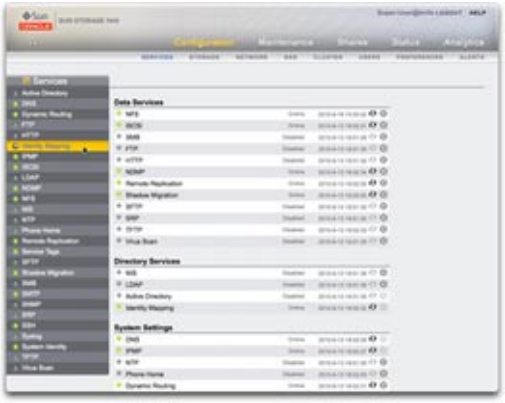
The detached storage controller, if any, can now be attached to its own storage, powered on, and configured normally. If you are replacing a failed storage controller, attach the replacement to the remaining storage controller and storage and begin the cluster setup task described above.

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.

◆ ◆ ◆ CHAPTER 4

Services

Services



The Services screen features a side panel for quick navigation between services.

Introduction

The following services may be configured on the appliance:

Data

Service	Description
NFS	Filesystem access via the NFSv3 and NFSv4 protocols

Service	Description
iSCSI	LUN access via the iSCSI protocol
SMB	Filesystem access via the SMB protocol
FTP	Filesystem access via the FTP protocol
HTTP	Filesystem access via the HTTP protocol
NDMP	NDMP host service
Remote Replication	Remote replication
Shadow Migration	Shadow data migration
SFTP	Filesystem access via the SFTP protocol
SRP	Block access via the SRP protocol
TFTP	Filesystem access via the TFTP protocol
Virus Scan	Filesystem virus scanning

Directory

Service	Description
NIS	Authenticate users and groups from a NIS service
LDAP	Authenticate users and groups from a LDAP directory
Active Directory	Authenticate users with a Microsoft Active Directory Server
Identity Mapping	Map between windows entities and Unix IDs

System

Service	Description
DNS	Domain name service client
Dynamic Routing	RIP and RIPng dynamic routing protocols
IPMP	IP MultiPathing for IP fail-over
NTP	Network time protocol client
Phone Home	Product registration and support configuration
Service Tags	Product inventory support

Service	Description
SMTP	Configure outgoing mail server
SNMP	SNMP for sending traps on alerts and serving appliance status information
Syslog	Syslog Relay for sending syslog messages on alerts, and forwarding service syslog messages
System Identity	System name and location

Remote Access

Service	Description
SSH	SSH for CLI access

BUI

The BUI services page lists the services in the above groups, along with state information and buttons for administration. Double clicking a service line will take you to the service screen. The buttons are:

icon	description
	Go to service screen to configure properties and view logs. Appears on mouse-over
	Service is enabled and working normally
	Service is offline or disabled
	Service has a problem and requires operator attention
	Enable/disable service
	Restart service
	Enable/disable not available for this service
	Restart currently unavailable (enable the service first)

See the Basic Usage section of the User Interface guide for the full reference of these icons.

Selecting a Service

To go to a service screen, click the status icon on the left - which will change to an arrow icon on mouse over. Service screens allow service properties to be configured.

A side panel of all services can be revealed by clicking the icon on the left of the left-most "Services" title. Reclicking this icon will hide the panel.

Enabling a Service

If the service is not online, click the power icon and the service should come online

Disabling a Service

If the service is online, click the power icon and the service should go offline

Setting Properties

Properties can be set by changing them in the BUI and then clicking "APPLY". The "REVERT" button will reset the properties to their previous state, before editing.

Viewing Service Logs

Some service screens also provide service logs. These logs can provide information to help diagnose service issues, including:

- Times when a service changed state
- Error messages from the service

Look to the top right for "Properties" and "Logs", click "Logs" to change to the log viewer. If "Logs" is not visible, the service does not provide logs.

The log content is custom to each individual service, and 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:

Example Log Message	Description
Executing start method	The service is starting up
Method "start" exited with status 0	The service reported a successful start (0 == success)
Method "refresh" exited with status 0	The service successfully refreshed its configuration based on its service settings
Executing stop method	The service is being shut down
Enabled	The service state was checked to see if it should be started (such as during system boot), and it was found to be in the enabled state

Example Log Message	Description
Disabled	The service state was checked to see if it should be started (such as during system boot), and it was found to be in the disabled state

This is an example from the NTP service:

```
[ Oct 11 21:05:31 Enabled. ]
[ Oct 11 21:07:37 Executing start method (...). ]
[ Oct 11 21:13:38 Method "start" exited with status 0. ]
```

The system was booted at 21:05, and there is an event in the log to show that this service was found to be enabled. At 21:07:37 this service began startup, which completed at 21:13:38 - some six minutes later. Due to the nature of NTP and system clock adjustment, this service can take minutes to complete start up, as shown by the log.

CLI

The CLI services section is under `configuration services`. The `show` command shows the current state of all services:

```
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
                             scrk => disabled
                             sftp => disabled
                           shadow => online
                             smtp => online
                             snmp => disabled
                              ssh => online
                           syslog => disabled
                             tags => online
                             tftp => disabled
                            vscan => disabled
```

```
Children:
                              ad => Configure Active Directory
                             smb => Configure SMB
                             dns => Configure DNS
                      dynrouting => Configure Dynamic Routing
                             ftp => Configure FTP
                            http => Configure HTTP
                        identity => Configure System Identity
                           idmap => Configure Identity Mapping
                            ipmp => Configure IPMP
                           iscsi => Configure iSCSI
                            ldap => Configure LDAP
                            ndmp => Configure NDMP
                             nfs => Configure NFS
                             nis => Configure NIS
                             ntp => Configure NTP
                     replication => Configure Remote Replication
                            scrk => Configure Phone Home
                            sftp => Configure SFTP
                          shadow => Configure Shadow Migration
                            smtp => Configure SMTP
                            snmp => Configure SNMP
                             srp => Configure SRP
                             ssh => Configure SSH
                          syslog => Configure Syslog
                            tags => Configure Service Tags
                            tftp => Configure TFTP
                           vscan => Configure Virus Scan
                         routing => Configure Routing Table
```

Selecting a Service

Select a service by entering its name. For example, to select `nis`:

```
caji:configuration services> nis
caji:configuration services nis>
```

Once selected, its state can be viewed, it can be enabled and disabled, and properties may be set.

Viewing Service State

Service state can be viewed using the `show` command:

```
caji:configuration services nis> show
Properties:
                       <status> = online
                          domain = fishworks
                       broadcast = true
                       ypservers =
caji:configuration services nis>
```

Enabling a Service

Use the `enable` command:

```
caji:configuration services nis> enable
```

Disabling a Service

Use the `disable` command:

```
caji:configuration services nis> disable
```

Setting Properties

Properties can be changed by using the `set` command. After setting the properties to the desired values, use `commit` to save and activate the configuration:

```
caji:configuration services nis> set domain="mydomain"
                        domain = mydomain (uncommitted)
caji:configuration services nis> commit
caji:configuration services nis> show
Properties:
                      <status> = online
                        domain = mydomain
                     broadcast = true
                     ypservers =
```

Property names are similar to those shown in the BUI, but usually shorter and sometimes abbreviated.

Viewing Service Logs

Service logs cannot currently be viewed from the CLI.

Service Help

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

   done                 => Finish operating on "nis"

   enable               => Enable the nis service

   disable              => Disable the nis 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.
```

NFS

Introduction

NFS (Network File System) is an industry standard protocol to share files over a network. NFS versions 2, 3, and 4 are supported. For more information on how the filesystem namespace is constructed, see the filesystem namespace section.

Properties

Property	Description
Minimum supported version	Controls which versions of NFS are supported
Maximum supported version	Controls which versions of NFS are supported
Maximum # of server threads	Maximum number of concurrent NFS requests. This should at least cover the number of concurrent NFS clients that is anticipated. Allowed range is 20 to 1000
Grace period	Seconds that all clients have to reclaim locks after an appliance reboot. During this period, the NFS service only processes reclaims of old locks. All other requests for service must wait until the grace period is over, which by default is 90. Reducing this value allows NFS clients to resume operation more quickly after a server reboot, but it increases the probability that a client is not able to recover all its locks. Allowed range is 15 to 600
DNS domain for NFSv4 identity	Use DNS domain when mapping NFSv4 user and group identities.
Custom NFSv4 identity domain	Override the DNS domain with this string when mapping NFSv4 users and group identities.
Enable NFSv4 delegation	Enable NFSv4 delegation. Delegation allows clients to cache files locally and make modifications without contacting the server. This option is on by default and typically results in better performance, but in rare circumstances can cause problems. Disabling this setting should only be done after careful performance measurements of your particular workload and validation that the setting has a measurable performance benefit. This option only affects NFSv4 mounts.

Property	Description
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, the realm name is the same as your DNS domain name, except that the realm name is in uppercase. This convention helps differentiate problems with the Kerberos service from problems with the DNS namespace, while using a name that is familiar.
Kerberos master KDC	Each realm 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 can handle database administration requests. For instance, changing a password or adding a new principal must be done on the master KDC.
Kerberos slave KDC	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	Identifies the client. By convention, a principal name is divided into three components: the primary, the instance, and the realm. A principal can be specified as `joe`, `joe/admin`, or `joe/admin@ENG.EXAMPLE.COM`.
Kerberos admin password	Password for admin principal.

Changing services properties is documented in the BUI and CLI sections of Services.

Setting the NFS minimum and maximum versions to the same value will cause 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 that NFS version with your workload), and wish to force clients to only use the version that works best.

Kerberos realms

Configuring a Kerberos realm will create certain service principals and add 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 the system is configured in a cluster, 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 will reset the password for each of those principals. If the system is already joined to an Active Directory domain, the system cannot be configured as part of a Kerberos realm.

For information on setting up KDCs and Kerberized clients, see http://docs.sun.com/app/docs/doc/816-4557/setup-8 (`http://docs.sun.com/app/docs/doc/816-4557/setup-8`)?a=view. After setting NFS properties for Kerberos, change the Security mode on the Shares > Filesystem > Protocols screen to a mode using Kerberos.

Logs

These logs are available for the NFS service:

Log	Description
network-nfs-server:default	Master NFS server log
appliance-kit-nfsconf:default	Log of appliance NFS configuration events
network-nfs-cbd:default	Log for the NFSv4 callback daemon
network-nfs-mapid:default	Log for the NFSv4 mapid daemon - which maps NFSv4 user and group credentials
network-nfs-status:default	Log for the NFS statd daemon - which assists crash and recovery functions for NFS locks
network-nfs-nlockmgr:default	Log for the NFS lockd daemon - which supports record locking operations for files

To view service logs, refer to the Logs section from Services.

Analytics

NFS activity can be monitored in detail in the Analytics section. This includes monitoring:

- NFS operations per second
- ... by type of operation (read/write/...)
- ... by share name
- ... by client hostname
- ... by accessed filename
- ... by access latency

and combinations of the above.

CLI

The following table describes the mapping between CLI properties and the BUI property descriptions above.

CLI Property	BUI Property
version_min	Minimum supported version
version_max	Maximum supported version
nfsd_servers	Maximum # of server threads
grace_period	Grace period
mapid_dns	DNS domain for NFSv4 identity
mapid_domain	Custom NFSv4 identity domain
enable_delegation	Enable NFSv4 delegation
krb5_realm	Kerberos Realm
krb5_kdc	Kerberos master KDC
krb5_kdc2	Kerberos slave KDC
krb5_admin	Kerberos admin principal

Tasks

NFS Tasks

▼ Sharing a filesystem over NFS

1. Go to Configuration->Services
2. Check that the NFS 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 NFS sharing is enabled. This screen also allows configuration of the NFS share mode (read/read+write).

iSCSI

Introduction

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.

Properties

Property	Description
Use iSNS	Whether iSNS discovery is enabled
iSNS Server	An iSNS server
Use RADIUS	Whether RADIUS is enabled
RADIUS Server	A RADIUS server
RADIUS Server Secret	The RADIUS server's secret

Changing services properties is documented in the BUI and CLI sections of services. The CLI property names are shorter versions of those listed above.

Authentication

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.

Authorization

The iSCSI service allows you to specify a global list of initiators that you can use within initiator groups.

Targets and Initiators

For more information on iSCSI targets and initiators, see the SAN section.

CLI

For examples of administering iSCSI initiators and targets, see the SAN section.

Tips

Troubleshooting

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.

SMB

Introduction

The SMB service provides access to filesystems using the SMB protocol. Filesystems must be configured to share using SMB from the Shares configuration.

Properties

Property	Description
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 Solaris Express Reference Manual Collection for *smb*.
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.
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 addreses 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.

Changing service properties is documented in the BUI and CLI sections of services. The CLI property names are shorter versions of those listed above.

Share Properties

Several share properties must be set in certain ways when exporting a share over SMB.

Property	Description
Case sensitivity	SMB clients expect case-insensitive behavior, so this property must be "mixed"' or "'insensitive".
Reject non UTF-8	If non-UTF-8 filenames are allowed in a filesystem, SMB clients may function incorrectly.
Non-Blocking Mandatory Locking	This property must be enabled to allow byte range locking to function correctly.
Resource name	The name by which clients refer to the share. For information about how this name is inherited from a project, see the Protocols documentation.
Share-level ACL	An ACL which adds another layer of access control beyond the ACLs stored in the filesystem. For more information on this property, see the Protocols documentation.

The case sensitivity and reject non UTF-8 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:

1. Configure the Active Directory service.
2. Establish an identity mapping strategy and configure the service.
3. Configure SMB.
4. Configure access control, ACL entries, and ACL inheritance on shares.

Note that 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.

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 allow the SMB local user `dfsadmin` to manipulate DFS namespaces.

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.

Autohome Rules

The autohome share feature eliminates the administrative task of defining and maintaining home directory shares for each user that accesses the system through the SMB protocol. Autohome rules map SMB clients to home directories. There are three kinds of autohome rules:

Type	Description
Name service switch	This autohome rule queries NIS or LDAP for a user's home directory, then exports that directory to the SMB client as its home directory.
All users	An autohome rule which finds home directories based on wildcard characters. When substituting for the user's name, "&" matches the user.
Particular user	An autohome rule which provides a home directory for a particular user.

A name service switch autohome rule and an autohome rule for all users cannot exist at the same time.

Local Groups

Local groups are groups of domain users which confer additional privileges to those users.

Group	Description
Administrators	Administrators can bypass file permissions to change the ownership on files.
Backup Operators	Backup Operators can bypass file access controls to backup and restore files.

MMC Integration

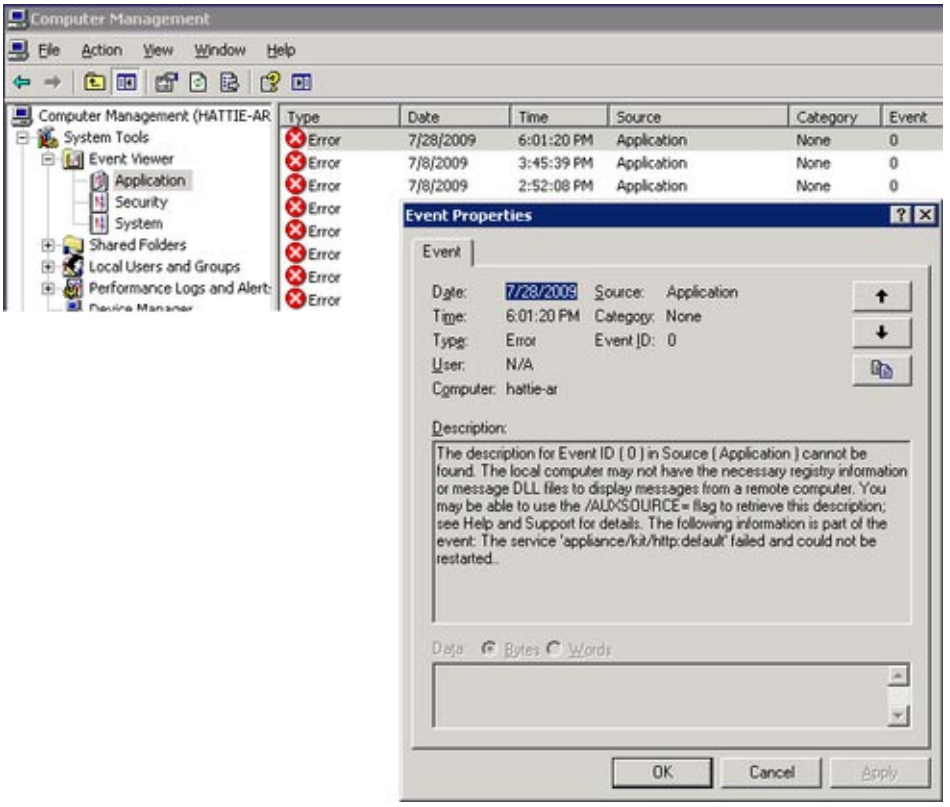
The Microsoft® Management Console (MMC) is an extensible framework of registered components, known as snap-ins, that provide comprehensive management features for both the local system and remote systems on the network. Computer Management is a collection of Microsoft Management Console tools, that may be used to configure, monitor and manage local and remote services and resources.

In order to use the MMC functionality on the Sun ZFS Storage 7000 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 Sun ZFS Storage 7000 appliances support the following Computer Management facilities:

Event Viewer

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 Sun ZFS Storage 7000 system. Following is a screen capture that illustrates the Application log and the properties dialog for an error event.



Share Management

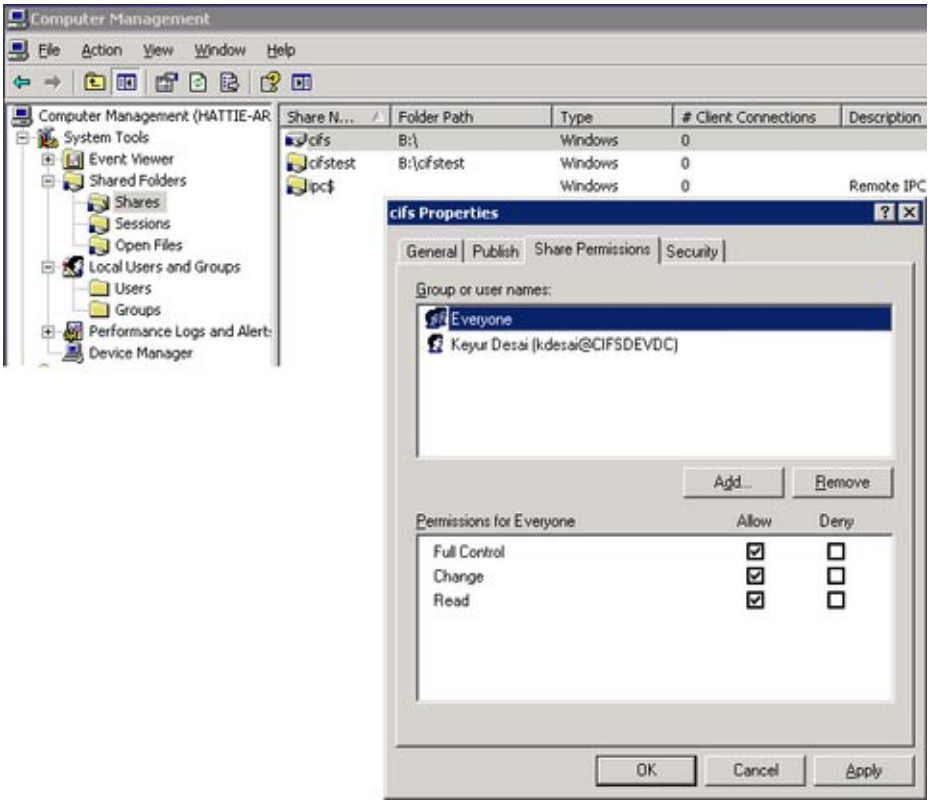
Support for share management includes the following:

- 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
- Setting client side caching property
- Setting maximum allowed or number of users property

Following is a screen capture that illustrates Permissions properties for a Share.

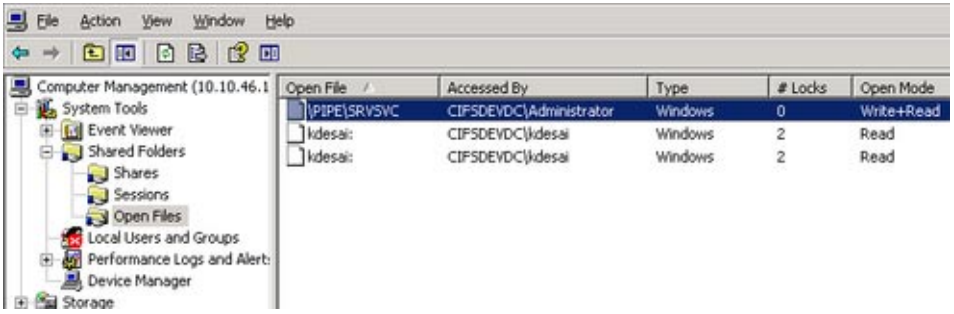


Users, Groups and Connections

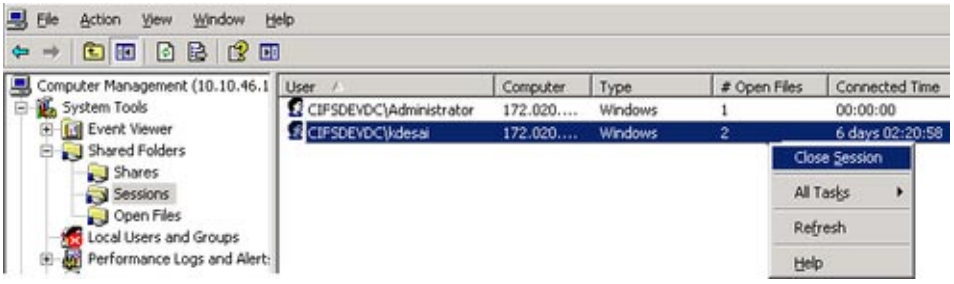
Supported features include the following:

- Viewing local SMB users and groups
- Listing user connections, including listing the number of open files per connection
- Closing user connections
- Listing open files, including listing the number of locks on the file and file open mode
- Closing open files

Following is a screen capture that illustrates open files per connection.

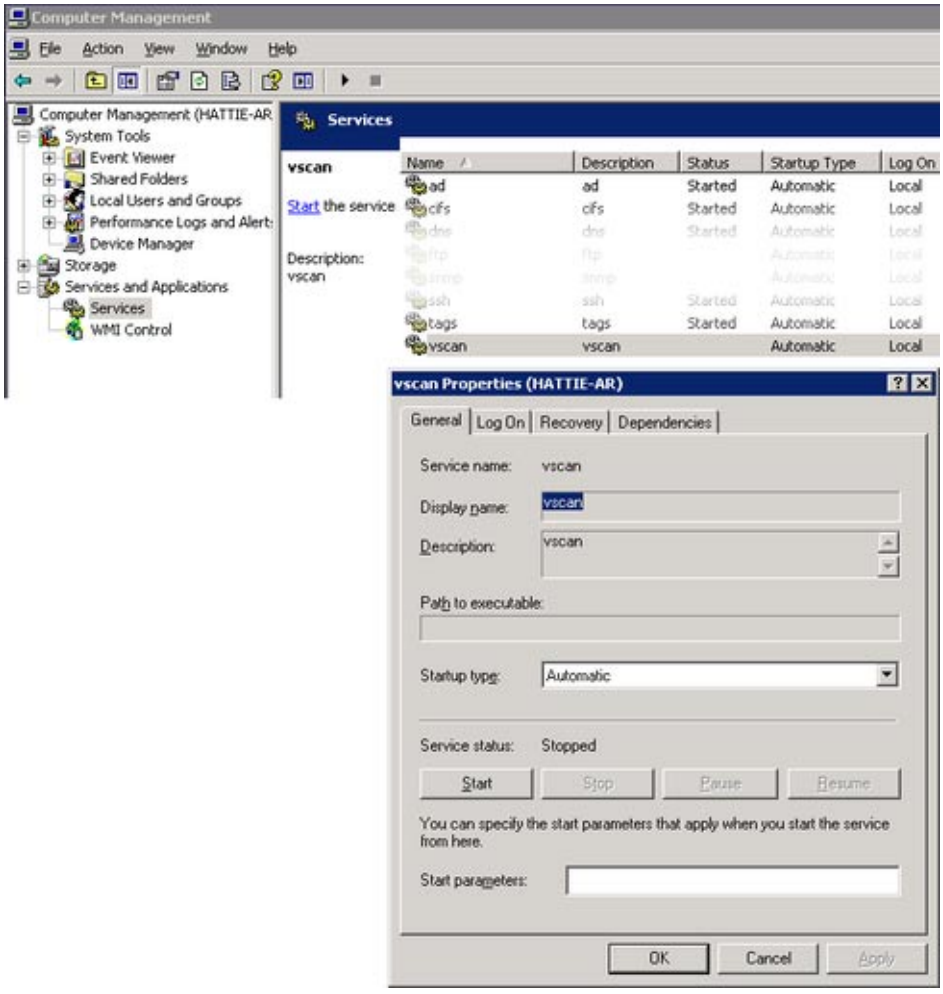


Following is a screen capture that illustrates open sessions.



Services

Support includes listing of services of the Sun ZFS Storage 7000 system. Services cannot be enabled or disabled using the Computer Management MMC application. Following is a screen capture that illustrates General properties for the vscan 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.

USERS	ALLOWED OPERATIONS
Regular users	List shares.
Members of the Administrators or Power Users groups	Manage shares, list user connections.

USERS	ALLOWED OPERATIONS
Members of the Administrators group	List open files and close files, disconnect user connections, view services and event log.

CLI

The following are examples of SMB administration at the CLI.

Adding autohome rules

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
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
```

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
```

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
```

Adding a user to a local group

```
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
```

Tasks

This section provides instructions for how to configure and enable the Sun ZFS Storage 7000 appliances for file sharing over SMB from initial configuration using the BUI.

SMB Tasks

▼ Initial Configuration

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 JBOD allocation for the storage pool. JBOD allocation may be none, half or all. If no JBODs are detected, check your JBOD 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.
23. Configure Remote Support.
24. If the appliance is not directly connected to the internet, configure an HTTP proxy through which the remote support service may communicate with Sun.
25. Enter your Sun 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.

▼ Active Directory Configuration

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.

▼ Project and Share Configuration

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.

▼ SMB Data Service Configuration

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.**

FTP

Introduction

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.

Properties

FTP Properties

General Settings

Property	Description
Port (for incoming connections)	The port FTP listens on. Default is 21
Maximum # of connections ("0" for unlimited)	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
Turn on delay engine to prevent timing attacks	This inserts small delays during authentication to fool attempts at user name guessing via timing measurements. Turning this on will improve security
Default login root	The FTP login location. The default is "/" and points to the top of the shares hierarchy. All users will be logged into this location after successfully authenticating with the FTP service

Property	Description
Logging level	The verbosity of the proftpd log.
Permissions to mask from newly created files and dirs	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

Security Settings

Property	Description
Enable SSL/TLS	Allow SSL/TLS encrypted FTP connections. This will ensure that the FTP transaction is encrypted. Default is disabled.
Port for incoming SSL/TLS connections	The port that the SSL/TLS encrypted FTP service listens on. Default is 21.
Permit root login	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
Maximum # of allowable login attempts	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

Changing services properties is documented in the BUI and CLI sections of Services. The CLI property names are shorter versions of those listed above.

Logs

Log	Description
proftpd	Logs FTP events, including successful logins and unsuccessful login attempts
proftpd_xfer	File transfer log
proftpd_tls	Logs FTP events related to SSL/TLS encryption

To view service logs, refer to the Logs section from Services.

Tasks

FTP Tasks

▼ Allowing FTP access to a share

1. **Go to Configuration->Services**
2. **Check 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.**

HTTP

Introduction

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.

Properties

Property	Description
Require client login	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 "nobody". See the section on authentication below.
Protocols	Select which access methods to support HTTP, HTTPS, or both.

Property	Description
HTTP Port (for incoming connections)	HTTP port, default is 80
HTTPS Port (for incoming secure connections)	HTTP port, default is 443

Changing services properties is documented in the BUI and CLI sections of services.

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).

Logs

Log	Description
network-http:apache22	HTTP service log

To view service logs, refer to the Logs section from Services.

Tasks

HTTP Tasks

▼ Allowing HTTP access to a share

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.**

NDMP

Introduction

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. Use the [[Maintenance:System:ConfigurationBackup|Configuration Backup/Restore]] feature for that.

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.

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":

Backup type	Details
dump	File-based for filesystems only. Supports file history and direct access recovery (DAR).
tar	File-based for filesystems only. Supports file history and direct access recovery (DAR).
zfs	Share-based for both filesystems and volumes. Does not support file history or direct access recovery (DAR), but may be faster for some datasets. Only supported with NDMPv4.

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.

Backing 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.

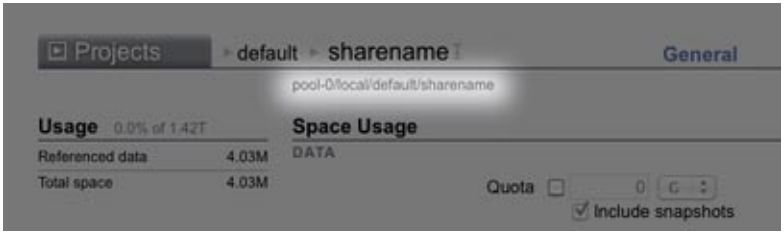
When restoring, if the destination of the restore path is not contained inside an existing share, projects and shares in the backup stream will be recreated as needed with their original properties as stored in the backup. For example, if you back up **/export/foo**, which contains project **proj1** 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.

Backing 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:



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 examples:

Canonical name	Shares backed up
pool-0/local/default	New snapshot of the local project called "default" and all of its shares.
pool-0/local/default@yesterday	Named snapshot "yesterday" of local project "default", and all of its shares having snapshot "yesterday".
pool-0/local/default/code	New snapshot of share "code" in local project "default". "code" could be a filesysten volume.
pool-0/local/default/code@yesterday	Named snapshot "yesterday" of share "code" in local project "default". "code" coul a filesystem or volume.

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.

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:

Variable	Details
LEVEL	Integer from 0 to 9 identifying the backup level.
DMP_NAME	Specifies a particular incremental backup set. Multiple sets of level incremental backups can be used concurrently by specifying different values for DMP_NAME.
UPDATE	Indicates whether this backup can be used as the base for subsequent incremental backups

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:

Day	Details
First of month	Level-0 backup. Backup contains all files in the share.
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).

Properties

The NDMP service configuration consists of the following properties:

Property	Description
DMA username and password	Used to authenticate the DMA (Data Management Application)
Enable DAR	Enables the system to locate files by position rather than by sequential search during restore operations. Enabling this option reduces the time it takes to recover a small number of files from many tapes. You must specify this option at backup time in order to be able to recover individual files later

Property	Description
Ignore file metadata changes for incremental backups	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 "tar" and "dump" backups and is disabled by default
Restore full absolute path for partial restore (v3 only)	Specifies that when a file is restored, the complete absolute path to that file is also restored (instead of just the file itself). This option is disabled by default.
NDMP version	The version of NDMP that your DMA supports
TCP port	The NDMP default connection port is 10000. NDMPv3 always uses this port. NDMPv4 allows a different port if needed.
Default restore pool(s)	When doing a full restore using types "tar" or "dump", the system will re-create datasets if there is not already a share mounted at the target. Because the NDMP protocol specifies only the mountpoint, the system will by default choose a pool in which to recreate any projects and shares. On a system with multiple pools, this property allows the user to explicitly specify one or more pools. Multiple pools need only be specified in a cluster with active pools on each head, and it is the responsibility of the user to make sure that this list is kept in sync with any storage configuration changes. If none of the pools exist or are online, then the system will select a default pool at random.
Rollback datasets before restore (ZFS backups only)	Only applies to backups with type "zfs". 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 "ZFS_FORCE" environment variable (see "Incremental Backups" above) but this property can be used to override the DMA setting to always rollback these datasets or never roll them back. Not rolling them back will cause the restore to fail unless they have already been manually rolled back. This property is intended for use with DMAs that do not allow administrators to configure custom environment variables like ZFS_FORCE.
DMA tape mode (for locally attached drives)	Specifies whether the DMA expects SystemV or BSD semantics. The default is SystemV, which is recommended for most DMAs. This option is only applicable for locally attached tape drives exported via NDMP. Consult your DMA documentation for which mode your DMA expects. Changing this option only changes which devices are exported when the DMA scans for devices, so you will need to reconfigure the tape devices in your DMA after changing this setting.

Changing services properties is documented in the BUI and CLI sections of Services.

Logs

Log	Description
system-ndmpd:default	NDMP service log

To view service logs, refer to the Logs section from Services.

SFTP

Introduction

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.

Properties

Property	Description
Port (for incoming connections)	The port SFTP listens on. Default is 218
Permit root login	Allow SFTP logins for the root user. This is on by default, since SFTP authentication is encrypted and secure
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.

Changing services properties is documented in the BUI and CLI sections of Services. The CLI property names are shorter versions of those listed above.

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
```

Logs

Log	Description
network-sftp:default	Logs SFTP service events

To view service logs, refer to the Logs section from Services.

Tasks

SFTP Tasks

▼ Allowing SFTP access to a share

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.**

Virus Scan

Introduction

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.

Properties

Property	Description
Maximum file size to scan	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.
Allow access to files that exceed maximum file size	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.

Changing services properties is documented in the BUI and CLI sections of services. The CLI property names are shorter versions of those listed above.

File Extensions

This section allows control over which files are or are not scanned, based on filename pattern matching. The default value, "*", will cause all files to be scanned (impacting performance on all file access). It may suit your environment to scan only a subset of files deemed to pose the greatest risk.

For example, to scan all high-risk filename patterns, including zip files, but not files whose names match the pattern "data-archive*.zip", one might configure this setting as follows:

Action	Pattern
Scan	exe
Scan	com
Scan	bat
Scan	doc
Don't Scan	data-archive*.zip
Don't Scan	*
Scan	zip

Note that "Don't Scan *" is required to prevent scanning of all other file types not explicitly included in the scan list.

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.

Property	Description
Enable	Use this scan engine
Host	Hostname or IP address of the scan engine server
Maximum Connections	Maximum number of concurrent connections. Some scan engines operate better with connections limited to 8.
Port	Port for the scan engine

Logs

Log	Description
vscan	Log of the Virus Scan service

To view service logs, refer to the Logs section from Services.

Tasks

The following are example tasks. See the BUI and CLI sections for how these tasks apply to each interface method.

Virus Scan Tasks

▼ Configuring virus scanning for a share

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.**

NIS

Introduction

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 FTP and HTTP/WebDAV.
- NIS users can be granted privileges for appliance administration. The appliance supplements NIS information with its own privilege settings.

Properties

Property	Description
Domain	NIS domain to use
Server(s): Search using broadcast	The appliance will send a NIS broadcast to locate NIS servers for that domain
Server(s): Use listed servers	NIS server hostnames or IP addresses

Changing services properties is documented in the BUI and CLI sections of Services.

The appliance will connect to the first NIS server listed or found using broadcast, and switch to the next if it stops responding.

Logs

Log	Description
network-nis-client:default	NIS client service log
appliance-kit-nsswitch:default	Log of the appliance name service, through which NIS queries are made
system-identity:domain	Log of the appliance domainname configurator

To view service logs, refer to the Logs section from Services.

Tasks

The following are example tasks. See the BUI and CLI sections for how these tasks apply to each interface method.

NIS Tasks

▼ Adding an appliance administrator from NIS

If you have an existing user in NIS who would like to login using their NIS credentials and administer the appliance:

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 Users for adding authorizations to this user.**

LDAP

Introduction

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 login to FTP and HTTP/WebDAV.
- 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.

Properties

Consult your LDAP server administrator for the appropriate settings for your environment.

Property	Description
Protect LDAP traffic with SSL/TLS	Use TLS (Transport Layer Security, the descendant of SSL) to establish secure connections to the LDAP server
Base search DN	Distinguished name of the base object, the starting point for directory searches.
Search scope	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 that the user's DN and password are not sent in plaintext. 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. "Self" authenticates the appliance using local authentication. Self authentication can only be used with the SASL/GSSAPI authentication method.
Proxy DN	Distinguished name of account used for proxy authentication.
Proxy Password	Password for account used for proxy authentication.
Schema definition	Schema used by the appliance. This property allows administrators to override the default search descriptor, attribute mappings, and object class mappings for users and groups. See "Custom Mappings" below.
Servers	List of LDAP servers to use. If only one server is specified, the appliance will only use that one server and LDAP services will be unavailable if that server fails. If multiple servers are specified, any functioning server may be used at any time without preference. If any server fails, another server in the list will be used. LDAP services will remain available unless all specified servers fail.

Changing services properties is documented in the BUI and CLI sections of Services.

Custom Mappings

To lookup 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 below, but this can be customized for different environments. The base search DN used in the examples below is *dc=example,dc=com*:

Search descriptor	Default value	Example
users	ou=people,*base search DN*	ou=people,dc=example,dc=com
groups	ou=group,*base search DN*	ou=group,dc=example,dc=com

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 and group attributes and objects, choose the appropriate tab ("Users" or "Groups") 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.

Logs

Log	Description
appliance-kit-nsswitch:default	Log of the appliance name service, through which LDAP queries are made

To view service logs, refer to the Logs section from Services.

Tasks

The following are example tasks. See the BUI and CLI sections for how these tasks apply to each interface method.

LDAP Tasks

▼ Adding an appliance administrator from LDAP

If you have an existing user in LDAP who would like to login using their LDAP credentials and administer the appliance:

1. **Go to Configuration->Services->LDAP**
2. **Set the LDAP service properties.**
3. **Apply/commit the configuration.**
4. **Go to Configuration->Users**
5. **Add user with type "directory"**
6. **Set username to their LDAP username**
7. **Continue with the instructions in Users for adding authorizations to this user.**

Active Directory

Introduction

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 for more information on local users.

Properties

Join Domain

The following table describes properties associated with joining an Active Directory domain.

Property	Description
Active Directory Domain	An Active Directory domain
Administrative User	An AD user who has credentials to create a computer account in Active Directory
Administrative 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.

Join Workgroup

The following table describes the configurable property for joining a workgroup.

Property	Description
Windows Workgroup	A workgroup

Changing services properties is documented in the BUI and CLI sections of services. The CLI property names are shorter versions of those listed above.

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. 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.

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.

Windows Server 2008 Support

Windows Version	Supported Software Versions	Workarounds
Windows Server 2003	all	none
Windows Server 2008 SP1	2009.Q2 3.1 and earlier	Apply hotfix for KB957441 as needed. (See section B.)
Windows Server 2008 SP1	2009.Q2 4.0 and later	Must apply hotfix for KB951191; apply hotfix for KB957441 as needed. (See sections A and B.)
Windows Server 2008 SP2	2009.Q2 4.0 and later	See Section C.
Windows Server 2008 R2	2009.Q2 4.0 and later	See Section C.

Section A: Kerberos issue (KB951191)

As originally shipped the appliance could interoperate with a Windows Server 2008 SP1 domain controller but it relied on a software workaround. This workaround dealt with a Windows Server 2008 SP1 Kerberos issue which was subsequently fixed by KB951191 (http://support.microsoft.com/default.aspx/kb/951191 (`http://support.microsoft.com/default.aspx/kb/951191`)). This fix was also incorporated into the Windows Server 2008 SP2 and R2 release.

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.

Section B: NTLMv2 issue (KB957441)

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.

Section C: Note on NTLMv2

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.

BUI

Use the "JOIN DOMAIN" button to join a domain, and the "JOIN WORKGROUP" button to join a workgroup.

CLI

To demonstrate the CLI interface, the following example will view the existing configuration, join a workgroup, and then join a domain.

```
twofish:> configuration services ad
twofish:configuration services ad> show
Properties:
                    <status> = online
                        mode = domain
                      domain = eng.fishworks.com

Children:
                         domain => Join an Active Directory domain
                       workgroup => Join a Windows workgroup
```

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
Properties:
                    <status> = disabled
                        mode = workgroup
                   workgroup = WORKGROUP
```

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
Properties:
                    <status> = online
                lmauth_level = 4
                         pdc = 192.168.3.21
                    ads_site = sf
twofish:configuration services smb> done
```

Following is an example of joining the new domain after the properties are configured.

```
twofish:> configuration services ad
twofish:configuration services ad> domain
twofish:configuration services ad domain> set domain=fishworks.com
twofish:configuration services ad domain> set user=Administrator
twofish:configuration services ad domain> set password=*******
twofish:configuration services ad domain> set searchdomain=it.fishworks.com
twofish:configuration services ad domain> commit
twofish:configuration services ad domain> done
twofish:configuration services ad> show
Properties:
                    <status> = online
                        mode = domain
                      domain = fishworks.com
```

Tasks

See the BUI and CLI sections for how these tasks apply to each interface method.

Active Directory Tasks

▼ Joining a Domain

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 contoller'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 a Workgroup

1. **Configure the workgroup name.**
2. **Apply/commit the configuration.**

Identity Mapping

Concepts

The identity mapping services manages Windows and Unix user identities simultaneously by using both traditional Unix UIDs (and GIDs) and Windows SIDs. 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.

The identity mapping service creates and maintains a database of mappings between SIDs, UIDs, and GIDs. Three different mapping approaches are available, as described in the following table:

Identity Mapping Concepts

Mapping Modes

Method	Description
IDMU	Retrieve mapping information from a Active Directory database using IDMU properties
Directory-based mapping	Retrieve mapping information from a Active Directory or LDAP database
Rule-based mapping	Configure mappings with name-based mappings

Method	Description
Ephemeral mapping	Let the system create on-demand, temporary mappings

When IDMU mapping is enabled, that method takes precedence over all other mapping methods. If directory-based mapping is enabled, that mapping approach will take precedence over the other approaches. If directory-based mapping is not available, then the service will attempt to map an identity the name-based approach. If no name-based rule is available for a given identity, the service will fallback on creating an ephemeral 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.

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. These extra attributes associated with the object must be configured in the following properties.

Identity Mapping Directory-based Mapping

Properties

Property	Description
Directory-Based Mapping	Whether directory-based mapping should be enabled

Property	Description
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

Changing services properties is documented in the BUI and CLI sections of services. The CLI property names are shorter versions of those listed above.

For information on augmenting the Active Directory or the LDAP schemas, see the section Directory-Based Identity Mapping for Users and Groups (Task Map) in the Solaris CIFS Administration Guide on www.docs.sun.com.

Name-based Mapping

The name-based mapping approach involves creating various rules which map identities by name. These rules establish equivalences between Windows identities and Unix identities.

Identity Mapping Name-based Mapping

Name-based Mapping Rules

The following properties comprise a name-based rule.

Property	Description
Mapping type	Whether this mapping grants or denies credentials
Mapping direction	The mapping direction. A mapping may map credentials in both directions, only from Windows to Unix, or only from Unix to Windows
Windows domain	The Active Directory domain of the Windows identity
Windows entity	The name of the Windows identity
Unix entity	The name of the Unix identity
Unix type	The type of the Unix identity, either a user or a group

Case Sensitivity

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.

Mapping Persistence

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.

Domain-Wide Rules

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.

Deny Mappings

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.

Icon	Description
	Maps Windows identity to Unix identity, and Unix identity to Windows identity
	Maps Windows identity to Unix identity
	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.

Ephemeral Mapping

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".

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.

Testing Mappings

The Mappings tab in the BUI shows how various identities are mapped given the current set of rules. By specifying a Windows entity or Unix entity, the entity will be mapped to its corresponding identity on the opposite platform. The resulting information in the User Properties and Group Properties sections displays information about the mapping identity, including the source of the mapping.

Examples

Here is a example of adding two name-based rules in the CLI. The first 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
twofish:configuration services idmap> list
MAPPING      WINDOWS ENTITY                   DIRECTION    UNIX ENTITY
idmap-000    Bill@eng.fishworks.com        (U) ==          wdp (U)
```

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
twofish:configuration services idmap (uncommitted)> set winname=*
twofish:configuration services idmap (uncommitted)> set direction=win2unix
twofish:configuration services idmap (uncommitted)> set unixname=
twofish:configuration services idmap (uncommitted)> set 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)
idmap-001    *@guest.fishworks.com         (U) =>           "" (U)
```

Tasks

The following are example tasks. See the BUI and CLI sections for how these tasks apply to each interface method.

Identity Mapping Tasks

▼ Configuring Identity Mapping

1. **Join an Active Directory domain.**
2. **Configure directory-based mapping (optional).**
3. **Configure deny mappings.**
4. **Configure name-based mappings.**

▼ Adding a Name-Based Mapping

1. **Configure whether the mapping grants or denies credentials.**
2. **Configure the domain and name for the Windows identity.**
3. **Configure the direction of the mapping.**
4. **Configure the name and type for the Unix identity.**
5. **Apply/commit the configuration**

DNS

Introduction

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 audit 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. You will receive an error message if your DNS server(s) do not meet this requirement.

Properties

Property	Description
DNS Domain	Domain name to search first when performing partial hostname lookups
DNS Server(s)	One or more DNS servers. IP addresses must be used.
Allow IPv4 non-DNS resolution	IPv4 addresses may be resolved to hostnames, and hostnames to IPv4 addresses, using NIS and/or LDAP if configured and enabled.
Allow IPv6 non-DNS resolution	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.

Changing services properties is documented in the BUI and CLI sections of Services.

CLI

The CLI includes builtins 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
```

Logs

Log	Description
network-dns-client:default	Logs the DNS service events

To view service logs, refer to the Logs section from Services.

Active Directory and DNS

If you plan to use Active Directory, at least one of 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. Therefore, if you must use non-DNS resolution, best results will likely be achieved by disabling DNS (see next section) and using NIS or LDAP exclusively for host resolution. You can validate host resolution results using the 'getent' CLI command described above.

Use of these options is strongly discouraged.

DNS-Less Operation

If the appliance will be unable to access any DNS servers from its installed location in the network, you may elect to operate without DNS by supplying the server 127.0.0.1. Use of this mode is strongly discouraged; several features will not work correctly, including:

- 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.

These limitations may be partially mitigated by using an alternate host resolution service; see "Non-DNS Resolution" above.

IPMP

Introduction

IPMP (Internet Protocol Network Multipathing) allows multiple network interfaces to be grouped as one, for both improved network bandwidth and reliability (interface redundancy). Some properties can be configured in this section. For the configuration of network interfaces in IPMP groups, see the Network section.

Properties

Property	Description
Failure detection latency	Time for IPMP to declare a network interface has failed, and to fail over its IP addresses
Enable fail-back	Allow the service to resume connections to a repaired interface

Changing services properties is documented in the BUI and CLI sections of Services.

Logs

Log	Description
network-initial:default	Logs the network configuration process

To view service logs, refer to the Logs section from Services.

Tasks

To configure IPMP, enable this service and follow the instructions in the Network section.

NTP

Introduction

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.

Properties

Property	Description	Examples
multicast address	Enter a multicast address here for an NTP server to be located automatically	224.0.1.1
NTP server(s)	Enter one or more NTP servers (and their corresponding authentication keys, if any) for the appliance to contact directly	0.pool.ntp.org
NTP Authentication Keys	Enter one or more NTP authentication keys for the appliance to use when authenticating the validity of NTP servers. See the Authentication section below.	Auth key: 10, Type: ASCII, Private Key: S

Changing services properties is documented in the BUI and CLI sections of Services.

Validation

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.

Authentication

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:

Type	Description	Example
DES	A 64 bit hexadecimal number in DES format	0101010101010101
NTP	A 64 bit hexadecimal number in NTP format	8080808080808080
ASCII	A 1-to-8 character ASCII string	topsecret
MD5	A 1-to-8 character ASCII string, using the MD5 authentication scheme.	md5secret

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.

BUI

To add NTP authentication keys in the BUI, click on the plus icon and specify the key number, type and private value for the new key. After the key has been added, it will appear as an option next to each specified NTP server.

CLI

Under `configuration services ntp`, edit authorizations with the `authkey` command:

```
clownfish:configuration services ntp> authkey
clownfish:configuration services ntp authkey>
```

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)
```

```
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
clownfish:configuration services ntp authkey>
```

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)
clownfish:configuration services ntp> set serverkeys=1,1
                    serverkeys = 1,1 (uncommitted)
clownfish:configuration services ntp> commit
clownfish:configuration services ntp>
```

To authenticate the server "gefilte" with key 1, "carp" with key 2 and "dory" with key 3:

```
clownfish:configuration services ntp> set servers=gefilte,carp,dory
                      servers = gefilte,carp,dory (uncommitted)
clownfish:configuration services ntp> set serverkeys=1,2,3
                    serverkeys = 1,2,3 (uncommitted)
clownfish:configuration services ntp> commit
clownfish:configuration services ntp>
```

To authenticate the servers "gefilte" and "carp" with key 1, and to additionally have an unauthenticated NTP server "dory":

```
clownfish:configuration services ntp> set servers=gefilte,carp,dory
                      servers = gefilte,carp,dory (uncommitted)
clownfish:configuration services ntp> set serverkeys=1,1,0
                    serverkeys = 1,1,0 (uncommitted)
clownfish:configuration services ntp> commit
clownfish:configuration services ntp>
```

BUI Clock

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.

Tips

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.

Log	Description
network-ntp:default	Log for the NTP service

To view service logs, refer to the Logs section from Services.

Tasks

NTP Tasks

▼ BUI Clock Synchronization

This will set the appliance time to match the time of your browser.

1. **Disable the NTP service.**
2. **Click the "SYNC" button.**

Remote Replication

Introduction

The remote replication service facilitates replication of projects and shares to and from other Sun Storage 7000 appliances. This functionality is described in detail in the Replication documentation.

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). Details on managing remote replication can be found in the Replication documentation.

Dynamic Routing

RIP and RIPng Dynamic Routing Protocols

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. Accordingly, disabling this service will cause all dynamic routes to be removed from the routing table. RIP and RIPng listen on UDP ports 520 and 521 respectively.

Logs

Log	Description
network-routing-route:default	Logs RIP service events
network-routing-ripng:quagga	Logs RIPng service events

Phone Home

Introduction

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 Sun's inventory portal, through which you can manage your Sun gear. Registration is also a prerequisite for using the Phone Home service.
- The Phone Home service communicates with Sun support to provide:
- * Fault reporting - the system reports active problems to Sun 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.
- * Heartbeats - daily heartbeat messages are sent to Sun to indicate that the system is up and running. Sun 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 Sun describing current software and hardware versions and configuration as well as storage configuration. No user data or metadata is transmitted in these messages.

Sun Online Account

You need a valid Sun Online account user name and password to use the fault reporting and heartbeat features of the Phone Home service. You might already have one if you registered for an account with programs such as Java Developer ConnectionSM, Online Support Center (OSC), My Sun, SunSolveSM, and Sun Store.

For automated service response, it is also important to provide a *Technical Contact* to Sun Support Services. When a Service Request is created, Sun Support Services will telephone or email the Technical Contact to resolve the problem. You can use the Sun Member Support Center to specify the Technical Contact for your Sun products. You can also contact Sun Support Services or your Sun account team for assistance.

Properties

Changing service properties is documented in the BUI and CLI sections of services. The phone home service is known as `scrk` within the CLI.

Web Proxy

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 Sun. These proxy settings will also be used to upload support bundles. See System Maintenance for more details on support bundles.

Property	Description
Use proxy	Connect via a web proxy
Host/port	Web proxy hostname or IP address, and port
Username	Web proxy username
Password	Web proxy password

Registration

To register the appliance for the first time, you must provide a Sun Online Account and specify one of that account's inventory teams into which to register the appliance. Using the BUI:

1. Enter your Sun Online Account user name and password. A privacy statement will be displayed for your review. It can be viewed at any time later in both the BUI and CLI.
2. The appliance will validate the credentials and allow you to 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 '$'.
3. Commit your changes.

In the CLI, this process involves configuring several properties of the service:

1. Set `soa_id` and `soa_password` to the user name and password for your Sun Online Account, respectively.
2. Commit your changes.
3. Set `domain_name` to the name of the inventory team in which you wish to register the appliance.
4. Commit your changes.

Once registered, the appliance cannot be unregistered, but the registration can be changed.

- Click 'Change account...' to change the Sun Online Account used by the appliance. You can then select one of that account's inventory teams. Commit your changes.
- To use the same account but register in a different inventory team, use the drop-down box to select a different inventory team. Commit your changes.

Status

Property	Description
Last heartbeat sent at	Time last heartbeat was sent to Sun support

Service state

If the phone home service is enabled before a valid Sun Online account has been entered, it will appear in the maintenance state. You must enter a valid Sun Online account to use the phone home service.

Logs

There is a log of Phone Home events in Maintenance->Logs->Phone Home.

SNMP

Introduction

The SNMP (Simple Network Management Protocol) service provides two different functions on the appliance:

- Appliance status information can be served by SNMP. See MIBs.

- Alerts can be configured to send SNMP traps.

Both SNMP versions 1 and 2c are available when this service is enabled.

Properties

Property	Description
SNMP community name	This is the community string that SNMP clients must provide when connecting
Authorized network	The network which is allowed to query the SNMP server, in CIDR notation. To block all clients, use 127.0.0.1/8 (localhost only); to allow all clients, use 0.0.0.0/0
Appliance contact	The string served by MIB-II OID .1.3.6.1.2.1.1.4.0. Setting this to a person or department's name will help SNMP clients determine who is responsible for this appliance
Trap destinations	The hostnames or IP addresses for sending SNMP traps to. Custom SNMP traps can be configured in the Alerts section. Set this to 127.0.0.1

Changing services properties is documented in the BUI and CLI sections of services. The CLI property names are shorter versions of those listed above. After changing properties, restart the SNMP service.

The SNMP service also provides the MIB-II location string. This property is sourced from the System Identity configuration.

MIBs

If the SNMP services is online, authorized networks will have access to the following MIBs (Management Information Bases):

MIB	Purpose
.1.3.6.1.2.1.1	MIB-II system - generic system information, including hostname, contact and location
.1.3.6.1.2.1.2	MIB-II interfaces - network interface statistics
.1.3.6.1.2.1.4	MIB-II IP - Internet Protocol information, including IP addresses and route table
.1.3.6.1.4.1.42	Sun Enterprise MIB (SUN-MIB.mib.txt)
.1.3.6.1.4.1.42.2.195	Sun FM - fault management statistics (MIB file linked below)
.1.3.6.1.4.1.42.2.225	Sun AK - appliance information and statistics (MIB file linked below)

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:

OID	Contents
.1.3.6.1.4.1.42.2.195.1.1	Fault Management problems
.1.3.6.1.4.1.42.2.195.1.2	Fault Management fault events
.1.3.6.1.4.1.42.2.195.1.3	Fault Management module configuration
.1.3.6.1.4.1.42.2.195.1.5	Fault Management faulty resources

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:

OID	Contents
.1.3.6.1.4.1.42.2.225.1.4	General appliance info
.1.3.6.1.4.1.42.2.225.1.5	Cluster status
.1.3.6.1.4.1.42.2.225.1.6	Share status

See the MIB file linked above for the full descriptions.

Tasks

The following are example tasks for SNMP. See the BUI and CLI sections for how these tasks apply to each interface method.

SNMP Tasks

▼ Configuring SNMP to serve appliance status

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

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.**

SMTP

Introduction

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.

Properties

Property	Description
Send mail through smarthost	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.
Smarthost hostname	Outgoing mail server hostname.
Allow customized from address	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.
Custom from address	The From address to use for outbound email.

Changing services properties is documented in the BUI and CLI sections of Services.

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.

Logs

Log	Description
network-smtp:sendmail	Logs the SMTP service events
mail	Log of SMTP activity (including mails sent)

To view service logs, refer to the Logs section from Services.

Service Tags

Introduction

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 Sun service, allowing you to easily keep track of your Sun equipment and also expedite service calls. The service tags are enabled by default.

Properties

Property	Description
Discovery Port	UDP port used for service tag discovery. Default is 6481
Listener Port	TCP port used to query service tag data. Default is 6481

Changing services properties is documented in the BUI and CLI sections of services.

System Identity

Introduction

This service provides configuration for the system name and location. There may be a need to change these if the appliance is moved to a different network location, or repurposed.

Properties

Property	Description
System Name	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
System Location	A text string to describe the where the appliance is physically located. If SNMP is enabled, this will be exported as the *syslocation* string in MIB-II

Changing services properties is documented in the BUI and CLI sections of services.

Logs

Log	Description
system-identity:node	Logs the System Identity service events and errors

To view service logs, refer to the Logs section from Services.

SSH

Introduction

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.

Properties

Property	Description	Examples
Server key length	The number of bits in the ephemeral key.	768
Key regeneration interval	Ephemeral key regeneration interval, in seconds.	3600
Login grace period	The SSH connection will be disconnected after this many seconds if the client has failed to authenticate.	120
Permit root login	Allows the root user to login using SSH.	yes

Changing services properties is documented in the BUI and CLI sections of services. The CLI property names are similar to those listed above.

Logs

Log	Description
network-ssh:default	Log of the SSH service events and errors

To view service logs, refer to the Logs section from Services.

Tasks

The following are example tasks. See the BUI and CLI sections for how these tasks apply to each interface method.

SSH Tasks

▼ Disabling root SSH access

1. **Set permit root login to false.**
2. **Apply/commit the configuration.**

Shadow Migration

Introduction

The shadow migration service allows for automatic migration of data from external or internal sources. This functionality is described in great detail in the Shadow Migration shares documentation. 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.

Properties

Property	Description
Number of Threads	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).

Changing services properties is documented in the BUI and CLI sections of services.

Managing Shadow Migration

Complete information on how to manage shadow migration can be found in the Shadow Migration shares documentation.

Syslog

Introduction

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.
- 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.

Properties

Property	Description
Protocol Version	The version of the Syslog protocol to use, either Classic or Modern
Destinations	The list of destination IPv4 and IPv6 addresses to which messages are relayed.

Changing services properties is documented in the BUI and CLI sections of services. The CLI property names are shorter versions of those listed above. After changing properties, restart the Syslog service.

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.

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`**, a **`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.

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.

Facility	Tag Name	Description
daemon	ak	Generic tag for appliance subsystems. All alerts will be tagged **ak**, indicating a SUNW-MSG-ID follows.
daemon	idmap	Identity Mapping service for POSIX and Windows identity conversion.
daemon	smbd	SMB Data Protocol for accessing shares.

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.

Field Name	Description
SUNW-MSG-ID	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/.
TYPE	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.
VER	The version of this encoding format itself. This description corresponds to version "1" of the SUNW-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 Sun 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.
SOURCE	The subsystem within the appliance software that emitted the event. This field is for Sun Service use only.
REV	The internal revision of the subsystem. This field is for Sun Service use only.

Field Name	Description
EVENT-ID	The Universally Unique Identifier (UUID) associated with this event. Sun'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. Sun Service personnel can use the EVENT-ID to retrieve additional postmortem information associated with the problem that may help Sun 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:

Sun Severity	Syslog Level	Description
Minor	LOG_WARNING	A condition occurred that does not currently impair service, but the condi to be corrected before it becomes more severe.
Major	LOG_ERR	A condition occurred that does impair service but not seriously.
Critical	LOG_CRIT	A condition occurred that seriously impairs service and requires immedia correction.

Receiver Configuration Examples

Most operating systems include a syslog receiver, but some configuration steps may be required to turn it on. Some examples for common operating systems are shown below. Consult the documentation for your operating system or management software for specific details of syslog receiver configuration.

Configuring a Solaris Receiver

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:

```
# 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:

```
Aug 14 21:34:22 poptart.sf.fishpong.com poptart ak: SUNW-MSG-ID: AK-8000-LM, \
TYPE: alert, VER: 1, SEVERITY: Minor\nEVENT-TIME: Fri Aug 14 21:34:22 2009\n\
PLATFORM: i86pc, CSN: 12345678, HOSTNAME: poptart\n\
SOURCE: jsui.359, REV: 1.0\n\
EVENT-ID: 92dfeb39-6e15-e2d5-a7d9-dc3e221becea\n\
DESC: A test alert has been posted.\n\
AUTO-RESPONSE: None.\nIMPACT: None.\nREC-ACTION: None.
```

Configuring a Linux Receiver

Most Linux distributions include a bundled **sysklogd**(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:

```
Aug 12 22:03:15 192.168.1.105 poptart ak: SUNW-MSG-ID: AK-8000-LM, \
TYPE: alert, VER: 1, SEVERITY: Minor EVENT-TIME: Wed Aug 12 22:03:14 2009 \
PLATFORM: i86pc, CSN: 12345678, HOSTNAME: poptart SOURCE: jsui.3775, REV: 1.0 \
EVENT-ID: 9d40db07-8078-4b21-e64e-86e5cac90912 \
DESC: A test alert has been posted. AUTO-RESPONSE: None. IMPACT: None. \
REC-ACTION: None.
```

◆ ◆ ◆ CHAPTER 5

Shares

Shares

Editing general properties for a filesystem. Shares with similar characteristics can be grouped together as a Project.

Introduction

The storage appliance exports filesystems as shares, which are managed in this section of the appliance. Shares can be grouped into projects for common administrative purposes, including space management and common settings.

- Concepts - general information about organizing storage and managing share properties
- Shadow Migration - automatically migrate data locally or from remote servers

- Space Management - manage space use on a per-share or per-user basis with quotas and reservations
- Filesystem Namespace - information about how the filesystem namespace is managed and exported
- Shares - manage filesystems and LUNs
- General - manage general properties on shares
- Protocols - manage protocol (NFS, SMB, iSCSI, etc) settings
- Access - manage user-based access control on filesystems
- Snapshots - manage automatic and manual snapshots on shares
- Projects - manage projects
- General - manage general properties on projects
- Protocols - manage protocol settings on projects
- Access - manage user-based access control on projects
- Snapshots - manage automatic and manual snapshots on projects
- Replication - configure data replication to other appliances
- Replication - manage replication sources targeting this appliance
- Schema - define customized properties for use with shares and projects

Concepts

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 quotas and reservations.

While multiple storage pools are supported, this type of configuration is generally discouraged because it provides significant drawbacks as described in the storage configuration section. 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.

Projects

All filesystems and LUNs are grouped into projects. A project defines a common administrative control point for managing shares. All 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.

Shares

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.

Properties

All projects and shares have a number of associated properties. These properties fall into the following groups:

Property Type	Description
Inherited	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 mountpoint and SMB properties. When inherited, these properties concatenate the project setting with their own share name.

Property Type	Description
Read-only	These properties represent statistics about the project and share and cannot be changed. The most common properties of this type are space usage statistics.
Space Management	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.
Create time	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.
Project default	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.
Filesystem local	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.
LUN local	These properties apply only to LUNs and are not inherited. They cannot be set on projects.
Custom	These are user defined properties. For more information, see the schema section.

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` snapshot 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.

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.

Shadow Migration

Shadow Data 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 Sun Storage 7000 series of appliances has two integrated solutions for migrating data that addresses the most common use cases. The first, remote replication, is intended for replicating data between one or more appliances, and is covered separately. 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 Sun Storage 7000 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.

Traditional Data Migration

Traditional file migration typically works in one of two ways: repeated synchronization or external interposition.

Migration via synchronization

This method works by taking an active host X and migrating data to the new host Y while X remains active. Clients still read and write to the original host while this migration is underway. Once the data is initially migrated, incremental changes are repeatedly sent until the delta is

small enough to be sent within a single downtime window. At this point the original share is made read-only, the final delta is sent to the new host, and all clients are updated to point to the new location. The most common way of accomplishing this is through the rsync tool, though other integrated tools exist. This mechanism has several drawbacks:

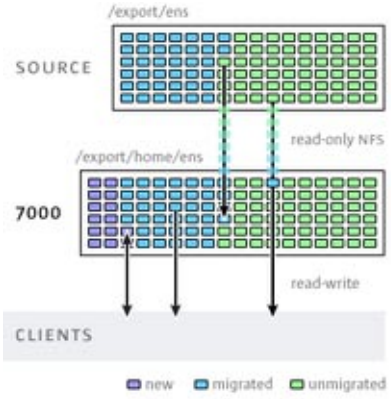
- The anticipated downtime, while small, is not easily quantified. If a user commits a large amount of change immediately before the scheduled downtime, this can increase the downtime window.
- During migration, the new server is idle. Since new servers typically come with new features or performance improvements, this represents a waste of resources during a potentially long migration period.
- Coordinating across multiple filesystems is burdensome. When migrating dozens or hundreds of filesystems, each migration will take a different amount of time, and downtime will have to be scheduled across the union of all filesystems.

Migration via external interposition

This method works by taking an active host X and inserting a new appliance M that migrates data to a new host Y. All clients are updated at once to point to M, and data is automatically migrated in the background. This provides more flexibility in migration options (for example, being able to migrate to a new server in the future without downtime), and leverages the new server for already migrated data, but also has significant drawbacks:

- The migration appliance represents a new physical machine, with associated costs (initial investment, support costs, power and cooling) and additional management overhead.
- The migration appliance represents a new point of failure within the system.
- The migration appliance interposes on already migrated data, incurring extra latency, often permanently. These appliances are typically left in place, though it would be possible to schedule another downtime window and decommission the migration appliance.

Shadow Migration



Shadow migration uses interposition, but is integrated into the appliance and doesn't 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 Sun Storage 7000 appliance. Clients can then access the appliance in read-write mode.

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 exists in two locations, which means that backups are more complicated, and snapshots may be incomplete and/or exist only on one host. Because of this, it is 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.

Shadow migration behavior

Restrictions on 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.

Shadow filesystem semantics during migration

If the client accesses a file or directory that has not yet been migrated, there is an observable effect on behavior:

- For directories, clients 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 affect.
- 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 singe filesystem, through the use of NFSv4 automatic client mounts (sometimes called "mirror mounts") or nested local mounts.

Identity and ACL migration

In order to properly migrate identity information for files, including ACLs, the following rules must be observed:

- 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.

Shadow Migration Management

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 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 progress

Monitoring progress of a shadow migration is difficult given the context in which the operation runs. A single filesystem can shadow all or part of a filesystem, or multiple filesystems with nested mountpoints. As such, there is no way to request statistics about the source and have any confidence in them being correct. In addition, even with migration of a single filesystem, the methods used to calculate the available size is not consistent across systems. For example, the remote filesystem may use compression, or it may or not include metadata overhead. For these reasons, it's impossible to display an accurate progress bar for any particular migration.

The appliance provides the following information that is guaranteed to be accurate:

- Local size of the local filesystem so far
- Logical size of the data copied so far
- Time spent migrating data so far

These values are made available in the BUI and CLI through both the standard filesystem properties as well as properties of the shadow migration node (or UI panel). If you know the size of the remote filesystem, you can use this to estimate progress. The size of the data copied consists only of plain file contents that needed to be migrated from the source. Directories, metadata, and extended attributes are not included in this calculation. While the size of the data

migrated so far includes only remotely migrated data, resuming background migration may traverse parts of the filesystem that have already been migrated. This can cause it to run fairly quickly while processing these initial directories, and slow down once it reaches portions of the filesystem that have not yet been migrate.

While there is no accurate measurement of progress, the appliance does attempt to make an estimation of remaining data based on the assumption of a relatively uniform directory tree. This estimate can range from fairly accurate to completely worthless depending on the dataset, and is for information purposes only. For example, one could have a relatively shallow filesystem tree but have large amounts of data in a single directory that is visited last. In this scenario, the migration will appear almost complete, and then rapidly drop to a very small percentage as this new tree is discovered. Conversely, if that large directory was processed first, then the estimate may assume that all other directories have a similarly large amount of data, and when it finds them mostly empty the estimate quickly rises from a small percentage to nearly complete. The best way to measure progress is to setup a test migration, let it run to completion, and use that value to estimate progress for filesystem of similar layout and size.

Canceling migration

Migration can be canceled, but 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 migrated 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.

Snapshots of shadow filesystems

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. This reflects the inconsistent state of the on-disk data during the migration.

Backing up shadow filesystems

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 filesystems

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.

Shadow migration analytics

In addition to standard monitoring on a per-share basis, it's also possible to monitor shadow migration system-wide through Analytics. The shadow migration analytics are available under the "Data Movement" category. There are two 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 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's 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.

Migration of local filesystems

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 can't be modified, such as creating a compressed version of a filesystem, or changing the recordsize 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.

Tasks

Testing potential shadow migration

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.

1. Configure the source so that the Sun Storage 7000 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 is 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

Once you are confident that the basic setup is functional, the shares can be setup for the final migration.

1. Schedule downtime during which clients can be quiesced and reconfigured to point to a new server.

2. Configure the source so that the Sun Storage 7000 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's 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 SS7000.

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.

Space Management

Introduction

The behavior of filesystems and LUNs with respect to managing physical storage is different on the 7000 series than on many other systems. As described in the Concepts page, 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.

Terms

Before getting into details, it is important to understand some basic terms used when talking about space usage on the appliance.

Space Management Terms

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 Sun Storage 7000 series, 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 X 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).

Filesystem 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.

Data quotas

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 the snapshot section, 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.

Data reservations

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. As described in the snapshot section, 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.

User and group settings

Viewing current usage

Regardless of whether user and group quotas are in use, current usage on a per-user or per-group basis can be queried for filesystems and projects. Storage pools created on older versions of software may need to apply deferred updates before making use of this feature. After applying the deferred update, it may take some time for all filesystems to be upgraded to a version that support per-user and per-group usage and quotas.

BUI

To view the current usage in the BUI, navigate to the "Shares -> General" page, under the "Space Usage -> Users and Groups" section. There you will find a text input with a dropdown type selection. This allows you to query the current usage of any given user or group, within a share or across a project. The following types are supported:

- **`User or Group`** - Search users or groups, with a preference for users in the case of a conflict. Since most user names don't overlap with group names, this should be sufficient for most queries.
- **`User`** - Search users.
- **`Group`** - Search groups.

Lookups are done as text is typed in the input. When the lookup is completed, the current usage will be displayed. In addition, the "Show All" link will bring up 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.

CLI

In the CLI, the `users` and `groups` commands can be used from the context of a particular project or share. From here, the `show` command can be used to display current usage in a tabular form. In addition, the usage for a particular user or group can be retrieved by selecting the particular user or group in question and issuing the `get` command.

```
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
```

User or group quotas

Quotas can be set on a user or group at the filesystem 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 only be applied to filesystems.
- 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.

BUI

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. Once you have finished entering the user or group name and the current usage is displayed, the quota can be set by checking the box next to "quota" and entering a value into the size field. To disable a quota, uncheck the box. Once any changes have been applied, click the 'Apply' button to make changes.

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. Once that error has been corrected, an attempt to apply the changes again will show the other error.

CLI

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 doesn't 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'.

```
clownfish:> shares select default select eschrock
clownfish:shares default/eschrock> users
clownfish:shares default/eschrock users> list
USER       NAME                            USAGE  QUOTA
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)
```

```
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
clownfish:shares default/eschrock users quota (uncomitted)> set name=bmc
                          name = bmc (uncommitted)
clownfish:shares default/eschrock users quota (uncomitted)> set quota=200G
                         quota = 200G (uncommitted)
clownfish:shares default/eschrock users quota (uncomitted)> commit
clownfish:shares default/eschrock users> list
USER       NAME                             USAGE  QUOTA
user-000   root                              321K      -
user-001   ahl                              9.94K      -
user-002   eschrock                         20.0G   100G
user-003   bmc                                  -   200G
```

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.

Filesystem Namespace

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.

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.

Protocol access to mountpoints

Regardless of protocol settings, every filesystem must have a mountpoint. However, the way in which these mountpoints are used depends on protocol.

NFSv2 / NFSv3

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

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 `/export/home` was shared, it is possible to mount `/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.

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.

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.

HTTP / HTTPS

Filesystems are exported under the `/shares` directory, so a filesystem at `/export/home` will appear at `/shares/export/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.

Shares

BUI

The Shares UI is accessed from "Shares -> Shares". The default view shows shares across all projects on the system.

List of Shares

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:

Field	Description
Name	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.
Size	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. See the usage statistics for more information.
Mountpoint	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.
GUID	The SCSI GUID for the LUN. See iSCSI for more information.

The following tools are available for each share:

Icon	Description
	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.
	Edit an individual share (also accessible by double-clicking the row).
	Destroy the share. You will be prompted to confirm this action, as it will destroy all data in the share and cannot be undone.

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:

- General
- Protocols
- Access
- Snapshots

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 [[Shares:Projects|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.

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.

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 the general properties section.

Total space

The sum of referenced data, snapshot data, and unused reservation.

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.

Compression ratio

If compression is enabled, this shows the compressions 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 the general properties section.

Case sensitivity

Controls whether directory lookups are case-sensitive or case-insensitive. It supports the following options:

BUI Value	CLI Value	Description
Mixed	mixed	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 "mangled" version of the conflicts that uniquely identify the filename.
Insensitive	insensitive	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 prevents clients from creating name conflicts that would cause mangled names to be used over SMB. This setting should only be used where SMB is the primary protocol and alternative protocols are considered second-class, where conformance to expected standards is not an issue.
Sensitive	sensitive	All lookups are case-sensitive, even over SMB where lookups are traditionally case-insensitive. In general, this setting should not be used because the SMB server can deal with name conflicts via mangled names, and may cause Windows applications to behave strangely.

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.

BUI Value	CLI Value	Description
None	none	No normalization is done.
Form C	formC	*Normalization Form Canonical Composition (NFC)* - Characters are decomposed and then recomposed by canonical equivalence.
Form D	formD	*Normalization Form Canonical Decomposition (NFD)* - Characters are decomposed by canonical equivalence.
Form KC	formKC	*Normalization Form Compatibility Composition (NFKC)* - Characters are decomposed by compatability equivalence, then recomposed by canonical equivalence.
Form KD	formKD	*Normalization Form Compatibility Decomposition (NFKD)* - Characters are decomposed by compatibility equivalence.

Volume block size

The native block size for LUNs. This can be any power of 2 from 512 bytes to 128K, 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 the section on Shadow Migration.

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.

Icon	Description
	Expand project panel
	Collapse project panel

Selecting a project from the panel will navigate to the project 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, see the Projects view.

Creating Shares

To create a share, view shares in a project or across all projects by selecting the "shares" sub-navigation entry. When selecting "Filesystems" or "LUNs," a plus icon will appear next to the name that will bring up a dialog to create the share. When creating a share, you can choose the target project from a pulldown menu, and provide a name for the share. The properties for each type of shares are defined elsewhere:

For Filesystems:

- User
- Group
- Permissions
- Mountpoint
- Reject non UTF-8 (create time only)
- Case sensitivity (create time only)
- Normalization (create time only)

For LUNs:

- Volume size
- Thin provisioned
- Volume block size (create time only)

CLI

The shares CLI is under `shares`

Navigation

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 = groupmask (inherited)
                         atime = true (inherited)
                casesensitivity = mixed
                       checksum = fletcher4 (inherited)
                    compression = off (inherited)
                  compressratio = 100
                         copies = 1 (inherited)
                       creation = Mon Oct 13 2009 05:21:33 GMT+0000 (UTC)
                     mountpoint = /export/foo (inherited)
                  normalization = none
                          quota = 0
                     quota_snap = true
                       readonly = false (inherited)
                     recordsize = 128K (inherited)
                    reservation = 0
                reservation_snap = true
                  secondarycache = all (inherited)
                          nbmand = false (inherited)
                       sharesmb = off (inherited)
                       sharenfs = on (inherited)
                        snapdir = hidden (inherited)
                       utf8only = true
                          vscan = false (inherited)
                       sharedav = off (inherited)
                       shareftp = off (inherited)
                     space_data = 43.9K
                space_unused_res = 0
                 space_snapshots = 0
                 space_available = 12.0T
                     space_total = 43.9K
                      root_group = other
                root_permissions = 700
                       root_user = nobody
```

Share Operations

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 = groupmask (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)
                        sharesmb = off (inherited)
                        sharenfs = on (inherited)
                         snapdir = hidden (inherited)
                           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 sharenfs=off
                        sharenfs = off (uncommitted)
clownfish:shares default/foo (uncommitted)> commit
clownfish:shares default>
```

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)
clownfish:shares default>
```

A share can be renamed from the project context using the `rename` command:

```
clownfish:shares default> rename foo bar
clownfish:shares default>
```

A share can be moved between projects from the project context using the `move` command:

```
clownfish:shares default> move foo home
clownfish:shares default>
```

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 section.

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.

CLI Name	Type	BUI Name	BUI Location
aclinherit	inherited	ACL inheritance behavior	Access
aclmode	inherited	ACL behavior on mode change	Access
atime	inherited	Update access time on read	General
casesensitivity	create time	Case sensitivity	Static
checksum	inherited	Checksum	General
compression	inherited	Data compression	General
compresratio	read-only	Compression ratio	Static
copies	inherited	Additional replication	General
creation	read-only	-	-
dedup	inherited	Data deduplication	General
exported	inherited, replication packages only	Export	General
initiatorgroup	LUN local	Initiator group	Protocols
logbias	inherited	Synchronous write bias	General
lunumber	LUN local	LU number	Protocols
lunguid	read-only, LUN local	GUID	Protocols
mountpoint	inherited	Mountpoint	General
nbmand	inherited	Non-blocking mandatory locking	General
nodestroy	inherited	Prevent destruction	General
normalization	create time	Normalization	Static
origin	read-only	Origin	Static
quota	space management	Quota	General

CLI Name	Type	BUI Name	BUI Location
quota_snap	space management	Quota / Include snapshots	General
readonly	inherited	Read-only	General
recordsize	inherited	Database record size	General
reservation	space management	Reservation	General
reservation_snap	space management	Reservation / Include snapshots	General
root_group	filesystem local	Group	Access
root_permissions	filesystem local	Permissions	Access
root_user	filesystem local	User	Access
rstchown	inherited	Restrict ownership change	General
secondary cache	inherited	Cache device usage	General
shadow	create time	Data Migration Source	Static
sharedav	inherited	Protocols / HTTP / Share mdoe	Protocols
shareftp	inherited	Protocols / FTP / Share mode	Protocols
sharenfs	inherited	Protocols / NFS / Share mode	Protocols
sharesmb	inherited	Protocols / SMB / Resource name	Protocols
snapdir	inherited	.zfs/snapshot visibility	Snapshots
space_available	read-only	Available space	Usage
space_data	read-only	Referenced data	Usage
space_snapshots	read-only	Snapshot data	Usage
space_total	read-only	Total space	Usage
space_unused_res	read-only	Unused reservation	Usage
sparse	LUN local	Thin provisioned	General
targetgroup	LUN local	Target group	Protocols
utf8only	create time	Reject non UTF-8	Static
volblocksize	create time	Volume block size	Static
vscan	inherited	Virus scan	General

General

General Share Properties

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.

For information on how these properties map to the CLI, see the Shares CLI section.

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. 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 manging space usage for filesystems, including quotas and reservations, see the Space Management section.

Volume size

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. See the Thin provisioned property for more information. 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.

Thin provisioned

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. For more information, see the Reservation property.

Properties

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.

Mountpoint

The location where the filesystem is mounted. This property is only valid for filesystems.

The following restrictions apply to the mountpoint property:

- Must be under `/export`.
- Cannot conflict with another share.
- Cannot conflict with another share on cluster peer to allow for proper failover.

When inheriting the mountpoint property, the current dataset name is appended to the project's mountpoint setting, joined with a slash ('/'). For example, if the "home" project has the mountpoint setting `/export/home`, then "home/bob" would inherit the mountpoint `/export/home/bob`.

SMB shares are exported via their resource name, and the mountpoint is not visible over the protocol. However, even SMB-only shares must have a valid unique mountpoint on the appliance.

Mountpoints can be nested underneath other shares, though this has some limitations. For more information, see the filesystem namespace section.

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.

Note that deduplication has no effect on the calculated size of a share, but does affect the amount of space used for the pool. For example, if two shares contain the same 1GB file, each will appear to be 1GB in size, but the total for the pool will be just 1GB and the deduplication ratio will be reported as 2x.

`Performance Warning`: by its nature, deduplication requires modifying the deduplication table when a block is written to or freed. If the deduplication table cannot fit in DRAM, writes and frees may induce significant random read activity where there was previously none. As a result, **`the performance impact of enabling deduplication can be severe.`** Moreover, for some

cases -- in particular, share or snapshot deletion -- the performance degradation from enabling deduplication may be felt pool-wide. In general, it is not advised to enable deduplication unless it is known that a share has a very high rate of duplicated data, and that that duplicated data plus the table to reference it can comfortably reside in DRAM. To determine if performance has been adversely affected by deduplication, enable advanced analytics and then use analytics to measure "ZFS DMU operations broken down by DMU object type" and check for a higher rate of sustained DDT operations (Data Duplication Table operations) as compared to ZFS operations. If this is happening, more I/O is for serving the deduplication table rather than file I/O.

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.

BUI value	CLI value	Description
Off	off	No compression is done
LZJB (Fastest)	lzjb	A simple run-length encoding that only works for sufficiently simple inputs, but doesn't consume much CPU.
GZIP-2 (Fast)	gzip-2	A lightweight version of the gzip compression algorithm.
GZIP (Default)	gzip	The standard gzip compression algorithm.
GZIP-9 (Best Compression)	gzip-9	Highest achievable compression using gzip. This consumes a significant amount of CPU and can often yield only marginal gains.

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).

BUI value	CLI value	Description
Fletcher 2 (Legacy)	fletcher2	16-bit fletcher checksum
Fletcher 4 (Standard)	fletcher4	32-bit fletcher checksum
SHA-256 (Extra Strong)	sha256	SHA-256 checksum

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 the storage configuration section. 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 disjoint 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. Note that this setting will change behavior even when no log devices are present, though the effects may be less dramatic.

BUI value	CLI value	Description
Latency	latency	Synchronous writes are optimized for latency, leveraging the dedicated log device(s), if any.
Throughput	throughput	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.

Database record size

Controls the block size used by the filesystem. This property is only valid for filesystems.

By default, filesystems will use a block size just large enough to hold the file, or 128K for large files. This means that any file over 128K in size will be using 128K blocks. If an application then writes to the file in small chunks, it will necessitate reading and writing out an entire 128K block, even if the amount of data being written is comparatively small.

Shares that host small random access workloads (i.e. databases) should tune this property to be approximately equal to the record size used by the database. In the absence of a precise number, 8K is typically a good choice for most database workloads. The property can be set to any power of 2 from 512 to 128K.

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.

BUI value	CLI value	Description
Normal (Single Copy)	1	Default behavior. Store a single copy of data blocks.
Two Copies	2	Store two copies of every data block.
Three Copies	3	Store three copies of every data block.

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 the Virus Scan section.

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 the schema section.

Protocols

Shares Protocols

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. The NFS, SMB, HTTP, and FTP properties apply only to filesystems, while the iSCSI properties apply only to LUNs.

In the BUI, each protocol shows the path by which clients using that protocol will refer to the share. For example, the filesystem "fs0" on the server "twofish" would be available at the following locations:

Protocol	Location
NFS	twofish:/export/fs0
SMB	\\twofish\fs0
HTTP	http://twofish/shares/export/fs0/ (`http://twofish/shares/export/fs0/`)
FTP	ftp://twofish/export/fs0/
SFTP	/export/fs0/

For iSCSI, initiators can discover the target through one of the mechanisms described in the SAN documentation.

NFS

BUI Property	CLI Property	Description
Share mode	off/ro/rw	Determines whether the share is available for reading only, for reading and writing, or neither. In the CLI, "on" is an alias for "rw".
Disable setuid/setgid file creation	nosuid	If this option is selected, clients will not be able to create files with the setuid (S_ISUID) and setgid (S_ISGID) bits set, nor to enable these bits on existing files via the chmod(2) system call.
Prevent clients from mounting subdirectories	nosub	If this option is selected, clients will be prevented from directly mounting subdirectories. They will be forced to mount the root of the share. Note: this only applies to the NFSv2 and NFSv3 protocols not to NFSv4.
Anonymous user mapping	anon	Unless the "root" option is in effect for a particular client, the root user on that client is treated as an unknown user, and all attempts by that user to access the share's files will be treated as attempts by a user with this uid. The file's access bits and ACLs will then be evaluated normally.
Character encoding	See below	Sets the character set default for all clients. For more information, see the section on character set encodings.
Security mode	See below	Sets the security mode for all clients.

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:

Type	CLI Prefix	Description	Example
Host(FQDN) or Netgroup	none	A single client whose IP address resolves to the specified fully-qualified name, or a netgroup containing fully-qualified names to which a client's IP address resolves	caji.sf.example.com
DNS Domain	.	All clients whose IP addresses resolve to a fully qualified name ending in this suffix	sf.example.com
Network	@	All clients whose IP addresses are within the specified IP subnet, expressed in CIDR notation	192.168.20.0/22

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. Management of netgroups can be complex; consider using IP subnet rules or DNS domain rules instead where possible.

CLI Considerations

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,

```
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"
```

Security Modes

Security modes are set on per-share basis and can have performance impact. The following table describes the Kerberos security settings.

Setting	Description
krb5	End-user authentication via Kerberos V5
krb5i	krb5 plus integrity protection (data packets are tamper proof)
krb5p	krb5 plus privacy (data packets cannot be snooped or otherwise examined by a third party)

krb5p cannot be used without also using krb5i and krb5, and krb5i cannot be used without also using krb5.

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:

- 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

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.

SMB

Property	Description
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.
Use ABE	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 the projects section 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.

SCSI

Property	Description
Target group	The targets over which this LUN is exported
Initiator group	The initiators which may access this LUN

Property	Description
LU (logical unit) number	As LUNs are associated with target and initiator groups, they are assigned unique logical unit numbers. This property controls whether a logical unit must have number zero, or whether its number can be automatically assigned. No two LUNs which share the same target group and initiator group may share a logical unit number.
Assigned LU number	The LU number assigned to this LUN.
Operational status	The operational status of this LUN. An offline LUN is inaccessible to initiators regardless of target or initiator configuration.
Fix LU number	A flag which fixes the LU number at its current value. When this flag is set, any change in target group or initiator group will not change the LU number, and any group change which creates a conflict will fail. When this flag is not set, any group change may reset the LU number to a value known not to cause a conflict.
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 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 which identifies the SCSI device. This GUID will remain consistent within different head nodes and replicated environments.

HTTP

Property	Description
Share mode	The HTTP share mode for this filesystem. One of none, read only, or read/write.

FTP

Property	Description
Share mode	The FTP share mode for this filesystem. One of none, read only, or read/write.

SFTP

Property	Description
Share mode	The SFTP share mode for this filesystem. One of none, read only, or read/write.

Access

Access Control

This view allows you to 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.

Root Directory Access

Controls basic acess 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 the Identity Mapping service. 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 the Identity Mapping service. 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.

Access type	Description
User	User that is the current owner of the directory.
Group	Group that is the current group of the directory.
Other	All other accesses.

For each access type, the following permissions can be granted.

Type		Description
Read	R	Permission to list the contents of the directory.
Write	W	Permission to create files in the directory.
Execute	X	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.

In the BUI, selecting permissions is done by click 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 the root directory ACL section below. 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.

ACL Behavior

For information on ACLs and how they work, see the root directory ACL documentation.

ACL behavior on mode change

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.

BUI Value	CLI Value	Description
Discard ACL	discard	All ACL entries that do not represent the mode of the directory or file are discarded.
Mask with user and group	groupmask	User and group permissions are reduced such that they are no greater than owner permission bits. This is the default behavior.
Do not change ACL	passthrough	No changes are made to the ACL other than generating the necessary ACL entries to represent the new mode of the file or directory.

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 only affect ACL entries that are flagged as inheritable - other entries are not propagated regardless of this property setting.

BUI Value	CLI Value	Description
Do not inherit entries	discard	No ACL entries are inherited. The file or directory is created according to the client and protocol being used.
Only inherit deny entries	noallow	Only inheritable ACL entries specifying "deny" permissions are inherited.
Inherit all but "write ACL" and "change owner"	restricted	Removes the "write_acl" and "write_owner" permissions when the ACL entry is inherited, but otherwise leaves inheritable ACL entries untouched. This is the default.
Inherit all entries	passthrough	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.
Inherit all but "execute" when not specified	passthrough-x	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 "passthrough" 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 "cc" or "gcc". 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.

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.

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.

Type	Description
Owner	Current owner of the directory. If the owner is changed, this ACE will apply to the new owner.
Group	Current group of the directory. If the group is changed, this ACE will apply to the new group.
Everyone	Any user.
Named User	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.
Named Group	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.

Mode	Description
Allow	The permissions are explicitly granted to the ACE target.
Deny	The permissions are explicitly denied to the ACE target.

	Permission	Description
	Read	
(r)	Read Data/List Directory	Permission to list the contents of a directory. When inherited by a file, permission to read the data of the file.
(x)	Execute File/Traverse Directory	Permission to traverse (lookup) entries in a directory. When inherited by a file, permission to execute the file.
(p)	Append Data/Add Subdirectory	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.
(a)	Read Attributes	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 `ls` and `stat` equivalents.
(R)	Read Extended Attributes	Permission to read the extended attributes of a file or do a lookup in the extended attributes directory.
	Write	
(w)	Write Data/Add File	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.
(d)	Delete	Permission to delete a file.
(D)	Delete Child	Permission to delete a file within a directory.
(A)	Write Attributes	Permission to change the times associated with a file or directory.
(W)	Write Extended Attributes	Permission to create extended attributes or write to the extended attributes directory.
	Admin	
(c)	Read ACL/Permissions	Permission to read the ACL.
(C)	Write ACL/Permissions	Permission to write the ACL or change the basic access modes.
(o)	Change Owner	Permission to change the owner.
	Inheritance	
(f)	Apply to Files	Inherit to all newly created files in a directory.
(d)	Apply to Directories	Inherit to all newly created directories in a directory.
(i)	Do not apply to self	The current ACE is not applied to the current directory, but does apply to children. This flag requires one of "Apply to Files" or "Apply to Directories" to be set.

	Permission	Description
(n)	Do not apply past children	The current ACE should only be inherited one level of the tree, to immediate children. This flag requires one of "Apply to Files" or "Apply to Directories" to be set.

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:

Type	Action	Access
Owner	Allow	Full Control
Group	Allow	Read and Execute
Everyone	Allow	Read and Execute

Snapshots

Introduction

Snapshots are read only copies of a filesystem at a given point of time. For more information on snapshots and how they work, see the concepts page.

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.

BUI Value	CLI Value	Description
Hidden	hidden	The .zfs directory is not visible when listing directory contents in the root of the filesystem. This is default.
Visible	visible	This .zfs directory appears like any other directory in the filesystem.

BUI

Listing Snapshots

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:

Field	Description
Name	The name of the snapshot. For manual snapshots, this is the name provided when the snapshot was created. Manual snapshots can be renamed by clicking on the name and entering a new value. For automatic snapshots, this is a name of the form ".auto-<timestamp>", and these snapshots cannot be renamed. Other forms of automatic snapshots may be created beginning with ".rr" or "bk-". These snapshots are used internally for remote replication and NDMP backup, and will be removed once the appropriate operation has been completed.
Creation	The date and time when the snapshot was created.
Unique	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.
Total	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.
Clones	Show the number of clones of the snapshot. When the mouse is over a snapshot row with a non-zero number of clones, a "Show..." link will appear. Clicking this link will bring up a dialog box that displays the complete list of all clones.

Taking Snapshots

To create a manual snapshot, click the  icon when the "Snapshots" tab is selected and the list of snapshots is shown. A dialog box will prompt for the snapshot name. Hitting the "apply" button will create the snapshot. There is no limit on the number of snapshots that can be taken, but each snapshot will consume some amount of resources (namely memory), so creating large numbers of snapshots can slow down the system, eventually grinding to a halt. The practical limit on the number of snapshots system-wide depends on the system configuration, but should be on the order of a hundred thousand or more.

Renaming a Snapshot

To rename a snapshot, click the name within the list of active snapshots. This will change to a text input box. After updating the name within the text input, hitting return or changing focus will commit the changes.

Destroying a Snapshot

To destroy a snapshot, click the icon when over the row for the target snapshot. 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

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.

To rollback a filesystem, click the icon for the destination snapshot. 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.

Cloning a Snapshot

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.

To create a clone, click the icon for the source snapshot. A dialog will prompt for the following values.

Property	Description
Project	Destination project. By default, clones are created within the current project, but they can also be created in different projects (or later moved between projects).
Name	The name to give to the clone.
Preserve Local Properties	By default, the all currently inherited properties of the filesystem will inherit from the destination project in the clone. Local settings are always preserved. Setting this property will cause any inherited properties to be preserved as local setting in the new clone.

Property	Description
Mountpoint	When preserving local properties, the clone must be given a different mountpoint, as shares cannot save the same mountpoint. This option is only available when "Preserve local properties" is set.

Scheduled Snapshots

In addition to manual snapshots, you can configure automatic snapshots according to an arbitrary schedule. These snapshots are named '.auto-<timestamp>', and can be taken on half hour, hourly, daily, weekly, or

monthly schedules. A schedule is a list of intervals and retention policies. To add a new interval, click the  icon when viewing the "Schedules" tab. Each interval has the following properties.

Property	Description
Frequency	One of "half hour", "hour", "day", "week", or "month". This indicates how often the snapshot is taken.
Offset	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 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.

Automatic snapshots can only 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.

CLI

To access the snapshots for a share, navigate to the share and run the `snapshots` command.

```
clownfish:> shares select default select builds
clownfish:shares default/builds> snapshots
clownfish:shares default/builds snapshots>
```

Listing Snapshots

Snapshots can be listed using the standard CLI commands.

```
clownfish:shares default/builds snapshots&gt' list
today
yesterday
clownfish:shares default/builds snapshots>
```

Taking Snapshots

To take a manual snapshot, use the `snapshot` command:

```
clownfish:shares default/builds snapshots> snapshot test
clownfish:shares default/builds snapshots>
```

Renaming a Snapshot

To rename a snapshot, use the `rename` command:

```
clownfish:shares default/builds snapshots> rename test test2
clownfish:shares default/builds snapshots>
```

Destroying a Snapshot

To destroy a snapshot, use the `destroy` command:

```
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>
```

You can also use the `destroy` command from the share context without selecting an individual snapshot:

```
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

To rollback to a snapshot, select the target snapshot and run the `rollback` command:

```
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.

Continue? (Y/N)
clownfish:shares default/builds@today>
```

Cloning a Snapshot

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 = groupmask (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>
```

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.

Scheduled Snapshots

Automatic scheduled snapshots can be configured using the `automatic` command from the snapshot context. 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.

```
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
automatic-000       day                 -                   14:30    7
clownfish:shares default/builds snapshots automatic> done
clownfish:shares default/builds snapshots>
```

Projects

BUI

The Projects UI is accesssed 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.

List of Projects

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:

Field	Description
Name	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.
Size	The total size of all shares within the project and unused reservation.

The following tools are available for each project:

Icon	Description
	Edit an individual project (also accessible by double-clicking the row).
	Destroy the project. You will be prompted to confirm this action, as it will destroy all data in the share and cannot be undone.

Editing a Project

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 complete set of functionality can be found in the section for each tab:

- General
- Protocols
- Access
- Snapshots

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.

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 the shares section.

Referenced data

Sum of all referenced data for all shares within the project, in addition to a small amount of project overhead. See the shares section 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 the shares section 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 the shares section 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.

Static Properties

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.

Compression ratio

See the shares section for a complete description.

Creating Projects

To create a project, view the list of projects and click the  button. Alternatively, the clicking the "Add..." button in the project panel will present the same dialog. Enter the project name and click apply to create the project.

CLI

The projects CLI is under `shares`

Navigation

To select a project, use the `select` command:

```
clownfish:> shares
clownfish:shares> select default
clownfish:shares default> get
                    aclinherit = restricted
                       aclmode = groupmask
                         atime = true
                      checksum = fletcher4
                   compression = off
```

```
                  compressratio = 100
                        copies = 1
                      creation = Thu Oct 23 2009 17:30:55 GMT+0000 (UTC)
                    mountpoint = /export
                         quota = 0
                      readonly = false
                    recordsize = 128K
                   reservation = 0
                 secondarycache = all
                          nbmand = false
                       sharesmb = off
                       sharenfs = on
                        snapdir = hidden
                          vscan = false
                       sharedav = off
                       shareftp = off
                  default_group = other
           default_permissions = 700
                default_sparse = false
                  default_user = nobody
          default_volblocksize = 8K
               default_volsize = 0
                    space_data = 43.9K
             space_unused_res = 0
      space_unused_res_shares = 0
              space_snapshots = 0
              space_available = 12.0T
                  space_total = 43.9K
clownfish:shares default>
```

Project Operations

A project is created using the `project` command. The properties can be modified as needed before committing the changes:

```
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)
                      shareftp = off (default)
                 default_group = other (default)
          default_permissions = 700 (default)
               default_sparse = true (default)
                 default_user = nobody (default)
          default_volblocksize = 8K (default)
               default_volsize = 0 (default)
                    aclinherit = (default)
                       aclmode = (default)
                         atime = (default)
                      checksum = (default)
                   compression = (default)
                        copies = (default)
                      readonly = (default)
```

```
                    recordsize = (default)
                secondarycache = (default)
                        nbmand = (default)
                       snapdir = (default)
                         vscan = (default)
                custom:contact = (default)
             custom:department = (default)
clownfish:shares home (uncommitted)> set sharenfs=off
                     sharenfs = off (uncommitted)
clownfish:shares home (uncommitted)> commit
clownfish:shares>
```

A project can be destroyed using the `destroy` command:

```
clownfish:shares> destroy home
This will destroy all data in "home"! Are you sure? (Y/N)
clownfish:shares>
```

This command can also be run from within the project context after selecting a project.

A project can be renamed using the `rename` command:

```
clownfish:shares> rename default home
clownfish:shares>
```

Selecting a pool in a cluster

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. You can change pools using the `set` command from the toplevel shares context:

```
clownfish:shares (pool-0)> set pool=pool-1
clownfish:shares (pool-1)>
```

Once the pool context has been select, projects and shares are managed within that pool using the standard CLI interfaces.

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.

CLI Name	Type	BUI Name	BUI Location
aclinherit	inherited	ACL inheritance behavior	Access
aclmode	inherited	ACL behavior on mode change	Access

CLI Name	Type	BUI Name	BUI Location
atime	inherited	Update access time on read	General
checksum	inherited	Checksum	General
compression	inherited	Data compression	General
compressratio	read-only	Compression ratio	Static
copies	inherited	Additional replication	General
creation	read-only	-	-
dedup	inherited	Data deduplication	General
default_group	creation default	Group	General
default_permissions	creation default	Permissions	General
default_sparse	creation default	Thin provisioned	General
default_user	creation default	User	General
default_volblocksize	creation default	Volume block size	General
default_volsize	creation default	Volume size	General
mountpoint	inherited	Mountpoint	General
nbmand	inherited	Non-blocking mandatory locking	General
quota	space management	Quota	General
readonly	inherited	Read-only	General
recordsize	inherited	Database record size	General
reservation	space management	Reservation	General
secondary cache	inherited	Cache device usage	General
sharedav	inherited	Protocols / HTTP / Share mdoe	Protocols
shareftp	inherited	Protocols / FTP / Share mode	Protocols
sharenfs	inherited	Protocols / NFS / Share mode	Protocols
sharesmb	inherited	Protocols / SMB / Resource name	Protocols
snapdir	inherited	.zfs/snapshot visibility	Snapshots

CLI Name	Type	BUI Name	BUI Location
space_available	read-only	Available space	Usage
space_data	read-only	Referenced data	Usage
space_snapshots	read-only	Snapshot data	Usage
space_total	read-only	Total space	Usage
space_unused_res	read-only	Unused reservation	Usage
space_unused_res_shares	read-only	Unused reservation of shares	Usage
vscan	inherited	Virus scan	General

General

General Project Properties

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 the Projects CLI section.

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 the concepts page.

Quota

Sets a maximum limit on the total amount of space consumed by all filesystems and LUNs within the project. For more information, see the shares section. Unlike filesystems, project quotas cannot exclude snapshots, and can only be enforced across all shares and their snapshots.

Reservation

Guarantees a minimum amount of space for use across all filesystems and LUNs within the project. For more information, see the shares section. Unlike filesystems, project reservation cannot exclude snapshots, and can only be enforcedc 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
- Read only
- Update access time on read
- Non-blocking mandatory locking
- Data compression
- Data deduplication
- Checksum
- Cache device usage
- Database record size
- Additional replication
- Virus scan

Custom Properties

Custom properties can be added as needed to attach user-defined tags to projects and shares. For more information, see the schema section.

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
- Group
- Permissions

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
- Thin provisioned
- Volume block size

Protocols

Project Protocols

Each project has protocol-specific properties which define the behavior of different protocols for that shares within that project. In general, shares inherit protocol-specific properties in a straightforward manner. Exceptions and special cases are noted here. For protocol issues, refer to Troubleshooting Protocols.

NFS

NFS share properties are inherited normally, and described in the shares documentation.

SMB

Property	Description
Resource name	The name by which SMB clients refer to this share.
Use ABE	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.

No two SMB shares on the same system may share the same resource name. When filesystems inherit resource names from a project, the share's resource name is constructed according to these rules:

Project's Resource Name	Share's Resource Name
"off"	The contained filesystems are not exported over SMB.
"on"	The contained filesystems are exported over SMB with their filesystem name as the resource name.
Anything other than "off" or "on"	A resource name of the form <*project's resource name*>_<*filesystem name*> is constructed for each filesystem.

iSCSI

iSCSI properties are not inherited.

HTTP

HTTP share properties are inherited normally, and described in the shares documentation.

FTP

FTP share properties are inherited normally, and described in the shares documentation.

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.

- ACL behavior on mode change
- ACL inheritance behavior

Snapshots

Introduction

Snapshots are read only copies of a filesystem at a given point of time. For more information on snapshots and how they work, see the concepts page. 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.

Snapshot Properites

.zfs/snapshot visible

The behavior of this property is identical to its behavior at the share level.

BUI

Project level snapshots are administered in the same way as share level snapshots. The following actions are documented under the shares section.

- Listing snapshots
- Taking snapshots
- Renaming a snapshot
- Destroying a snapshot
- Scheduled Snapshots

Project snapshots do not support rollback or clone operations.

CLI

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. The following actions are documented under the shares section.

- Listing snapshots
- Taking snapshots
- Renaming a snapshot
- Destroying a snapshot
- Scheduled Snapshots

Project snapshots do not support rollback or clone operations.

Replication

Remote Replication Introduction

Sun Storage 7000 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 datacenter), 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 7000 series appliances when upgrading hardware or rebalancing storage. Shadow migration can also be used for this purpose.

The remote replication feature has several important properties:

- **Snapshot-based**. The replication subsystem takes a snapshot as part of each update operation and sends either the entire project contents up to the snapshot in the case of a full update. In the case of 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, to allow sharing over NFS to a different set of hosts than on the source. See Manging Replication Targets for details.
- **Secure**. The replication control protocol used among Sun Storage 7000 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 Targets below.

Concepts

Terminology

- **replication peer** (or just **peer**, in this context): a Sun Storage 7000 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 Project-level vs. Share-level below.
- **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).

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 setup 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:

- a name (used only to identify the target in the source appliance's BUI and CLI)
- a network address or hostname (to contact the target appliance)
- the target appliance's root password (to authorize the administrator to setup 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 either 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.

To configure replication targets, see Configuring Replication below.

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 Project-level versus Share-level Replication below). 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 projects 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 (see below).

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.

One special case of this needs explicit mention. 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 for any reason after having replicated some data (thus leaving incomplete data inside the package), subsequent replication updates using the same action will fail because the appliance will not overwrite the already-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, factory reset caused all such 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 so will be available even after factory reset. However, target information will still be lost, and actions with missing targets currently cannot be configured to point to a new target.

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 or change the IP address associated with an action.

Project-level vs Share-level Replication

The appliance allows administrators to configure remote replication on both the project or 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 shares which override the project's configuration form a new group consisting of only the project and share themselves.

For example, suppose we have the following:

- a project `home` and shares `bill`, `cindi`, and `dave`.
- `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`.

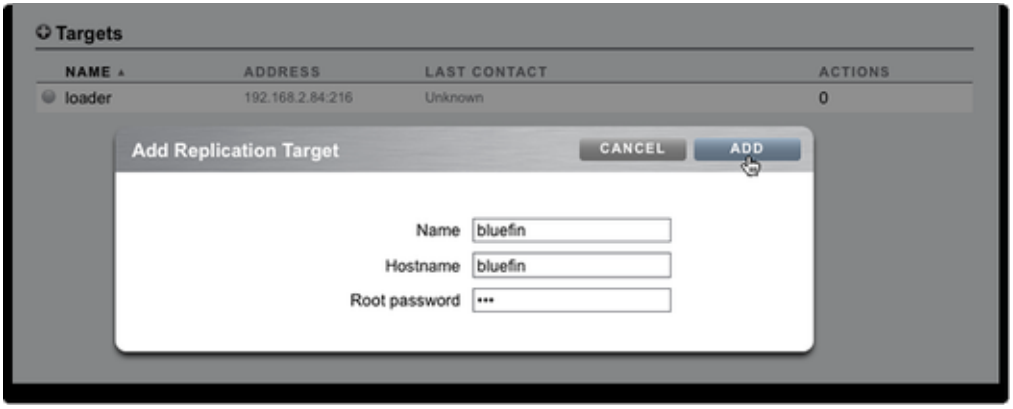
It is strongly recommended that project- and share-level replication be avoided within the same project because it can lead to surprising results (particularly when reversing the direction of replication). See the documentation for Managing Replication Packages for more details.

Configuring Replication

Be sure to read and understand the above sections on replication targets, actions, and packages before configuring replication.

Creating and Editing Targets

Targets are configured under Configuration > Services > Remote Replication. In the BUI, click the Targets tab:



In the CLI, navigate to the `targets` node to set or unset the target `hostname`, `root_password`, and `label` .

```
knife:> configuration services replication targets
```

From this context, administrators can:

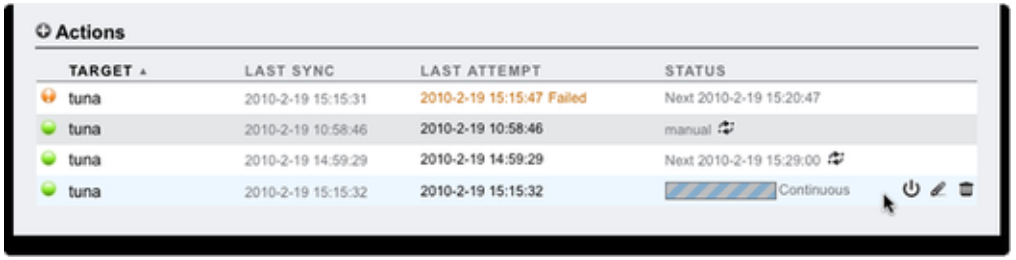
- Add new targets
- View the actions configured with the existing target
- Edit the unique identifier (label) for a target
- Destroy a target, if no actions are using it

Targets 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.

Creating and Editing Actions

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 and allow administrators to create new actions:



Replication actions have the following properties, which are presented slightly differently in the BUI and CLI:

Property (CLI name)	Description
Target	Unique identifier for the replication target system. This property is specified when an action is initially configured and immutable thereafter.

Property (CLI name)	Description
Pool	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.
Enabled	Whether the system will send updates for this action.
Mode (CLI: continuous) and schedule	Whether this action is being replicated continuously or at manual or scheduled intervals. See below for details.
Include Snapshots	Whether replication updates include non-replication snapshots. See below for details.
Limit bandwidth	Specifies a maximum speed for this replication update (in terms of data transferred over the network per second).
Use SSL	Whether to encrypt data on the wire using SSL. Using this feature can have a significant impact on per-action replication performance.
State	Read-only property describing whether the action is currently idle, sending an update, or cancelling an update.
Last sync	Read-only property describing the last time an update was successfully sent. This value may be unknown if the system has not sent a successful update since boot.
Last attempt	Read-only property describing the last time an update was attempted. This value may be unknown if the system has not attempted to send an update since boot.
Next update	Read-only property describing when the next attempt will be made. This value could be a date (for a scheduled update), "manual," or "continuous."

Modes: Manual, Scheduled, or Continuous

Replication actions can be configured to send updates manually, on a schedule, or continuously. The replication update process itself is the same in all cases. This property only controls the interval.

Because continuous replication actions send updates as frequently as possible, they essentially 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.

Note that continuous replication is still asynchronous. Sun 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.

Including Intermediate 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 Cancelling Updates

For targets that have been configured with scheduled or manual replication, administrators can choose to 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. Make sure 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 will display a barber-pole progress bar and the CLI will show 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.

Managing Replication Packages

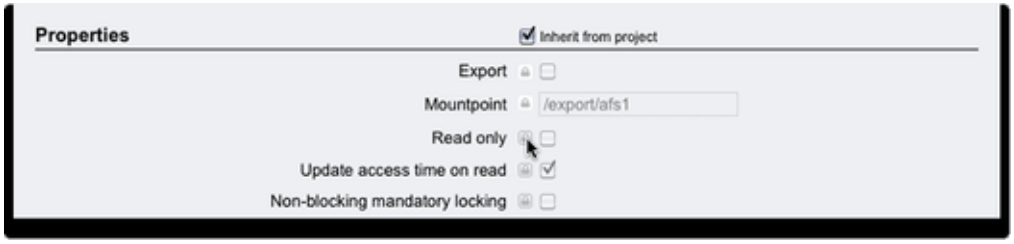
Packages are containers for replicated projects and shares. Each replication action on a source appliance corresponds to one package on the target appliance as described above. Both the BUI and CLI enable administrators to browse replicated projects, shares, snapshots, and properties much like local 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., sharenfs, 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 (see below). 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:



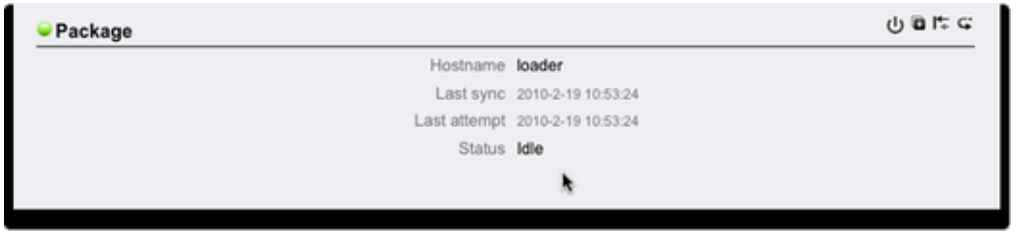
Note that the deferred updates provided with the 2010.Q1 release must be applied on replication targets in order to modify properties on such targets. The system will not allow administrators to modify properties inside replication packages on systems which have not applied the 2010.Q1 deferred updates.

Note that the current release does not support configuration of "chained" replication (that is, replicating replicated shares to another appliance).

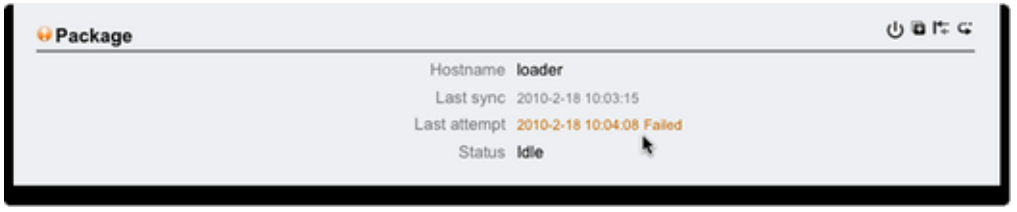
BUI

Replication packages are displayed in the BUI as projects under the "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 (see below):



The status icon on the left changes when replication has failed:



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.

CLI

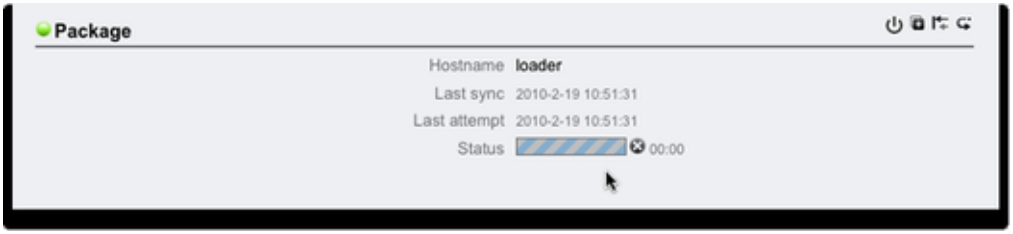
Replication packages are organized in the CLI by source under `shares replication sources`. Administrators first select a source, then a package. Package-level operations can be performed on this node (see below), or the project can be selected to manage project properties and shares just like local projects and shares with the exceptions described above. For example:

```
loader:> shares replication sources
loader:shares replication sources> show
Sources:

source-000 ayu
            PROJECT    STATE       LAST UPDATE
package-000 oldproj    idle        unknown
package-001 aproj1     receiving   Sun Feb 21 2010 22:04:35 GMT+0000 (UTC)

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
```

Cancelling 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:



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 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 Package or Individual Shares

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 (see Cloning a Snapshot). 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 `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 (see below). 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 `recv-XXX`).

Administrators can also clone individual replicated share snapshots using the normal BUI and CLI interfaces.

Exporting Replicated Filesystems

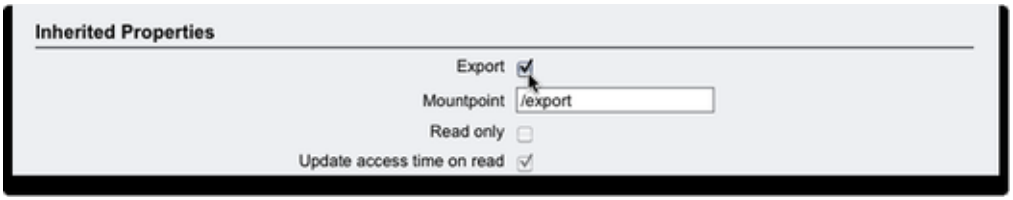
Replicated filesystems can be exported read-only to NAS clients. This can be used to verify the replicated data or to perform backups or other intensive operations on the replicated data (offloading such work from the source appliance).

The filesystem's contents always matches 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 "Snapshots and Data Consistency" below 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.

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 datacenter) 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:

1. The primary system is serving the production workload and replicating to the secondary system.
2. 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.
3. 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 direction of

replication is reversed 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 severed 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 mountpoint 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 mountpoints 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 mountpoint conflicts when the systems are first setup 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 `shares replication sources` 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.

Examples

Below is an example of cloning a received replication project, overriding both the project's and one share's mountpoint:

```
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 bob> 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

    SHARE                          MOUNTPOINT
    bob                            /export/bob (overridden)
    myfs1                          (inherited)
Are you sure you want to clone this project?
There are no conflicts.
perch:shares replication source-000 package-000 clone>
```

Remote Replication Details

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 authorizations under the "Projects and Shares" scope:

Authorization	Details
rrsource	Allows administrators to create, edit, and destroy replication targets and actions and send and cancel updates for replication actions.

Authorization	Details
rrtarget	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.

Note that 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 the Authorizations documentation.

Alerts

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).

Replication and Clustering

Replication can be configured from any SS7000 appliance to any other SS7000 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 an appliance to itself or its cluster peer is unsupported. Shadow migration can be used to copy data between storage pools (e.g., to rebalance storage) on a single appliance or in a cluster.
- 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.

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 the Clustering documentation.

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 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.

Snapshot Management

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:

1. 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.

2. Take a new project-level snapshot.
3. 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.
4. Record the new snapshot as the base snapshot for the next update and destroy the previous base snapshot (for incremental updates).

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. Note that snapshots may be created on shares not being sent as part of the update that are in the same project.
- 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).
- Relatedly, 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.

Replicating iSCSI Configuration

As described above, 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 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.

Observing Replication

While replication-specific analytics are not currently available, administrators can use the advanced TCP analytics to observe traffic by local port. Replication typically uses port 216 on the source appliance.

The status of individual replication actions and packages can be monitored using the BUI and CLI. See "Configuring Replication" above.

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:

Failure	Details
Cancelled	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.
Network connectivity failure	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.
Peer verification failed	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 Targets above.
Peer RPC failed	A remote procedure call failed on the target system. This occurs most commonly when the target appliance is running incompatible software. See "Migrating configuration from 2009.Q3 and earlier" below for more details.
No package	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.
Non-empty package exists	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 replicating some data. 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.
Disabled	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.
Target busy	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.
Out of space	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.

Failure	Details
Incompatible target	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.
Misc	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.

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 manual or scheduled updates fail, the system does not automatically try again until the next scheduled update (if any). When continuous replication fails, the system waits several minutes and tries again. The system will continue retrying failed continuous replications indefinitely.

When a replication update is in progress and another update is scheduled to occur, the latter update is skipped entirely rather than started immediately after the previous update completes. The next update will be sent only when the next update is scheduled to occur. The system posts an alert when an update is skipped for this reason.

Upgrading From 2009.Q3 and Earlier

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 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.

Schema

Customized Share Properties

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.

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:

Field	Description
NAME	The CLI name for this property. This must contain only alphanumeric characters or the characters ".:_\".
DESCRIPTION	The BUI name for this property. This can contain arbitrary characters and is used in the help section of the CLI

Field	Description
TYPE	The property type, for validation purposes. This must be one of the types described below.

The valid types for properties are the following

BUI Type	CLI Type	Description
String	String	Arbitrary string data. This is the equivalent of no validation.
Integer	Integer	A positive or negative integer
Positive Integer	PositiveInteger	A positive integer
Boolean	Boolean	A true/false value. In the BUI this is presented as a checkbox, while in the CLI it must be one of the values "true" or "false".
Email Address	EmailAddress	An email address. Only minimal syntactic validation is done.
Hostname or IP	Host	A valid DNS hostname or IP (v4 or v6) address.

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.

CLI

The schema context can be found at "shares -> schema"

```
carp:> shares schema
carp:shares schema> show
Properties:

NAME          TYPE          DESCRIPTION
owner         EmailAddress  Owner Contact
```

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>":

```
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
```

Tasks

Create a property to track contact info

In 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 the "Apply" button
7. Navigate to an existing share or project
8. Change the "Owner Contact" property under the "Custom Properties" section.

In the CLI:

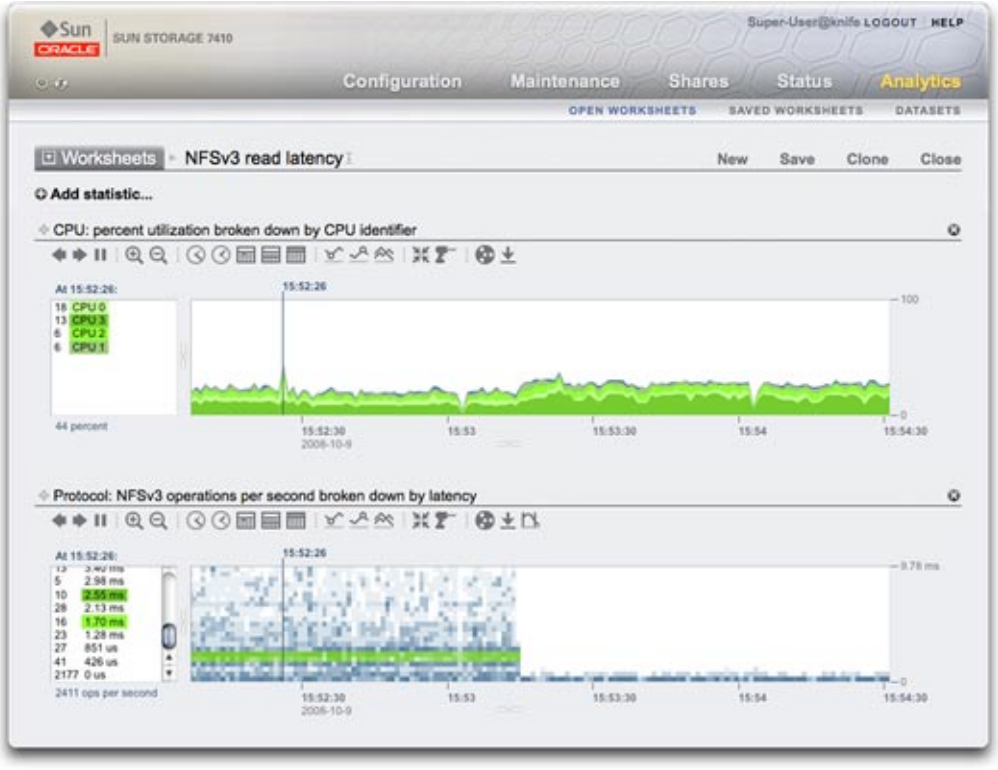
1. Navigate to the schema context (`shares schema`)

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

◆ ◆ ◆ CHAPTER 6

Analytics

Analytics



Using analytics to examine CPU utilization and NFSv3 operation latency

Introduction

This appliance is equipped with an advanced DTrace based facility for server analytics. Analytics provides real time graphs of various statistics, which can be saved for later viewing. About a dozen high level statistics are provided, such as NFSv3 operations/sec, which can then be customized to provide lower level details. Groups of viewed statistics can be saved as worksheets for future reference. To learn about the interface for Analytics, see Open Worksheets.

- Concepts - analytics overview
- Statistics - about the available statistics
- Overhead - performance overhead of statistics
- Open Worksheets - the main page for viewing analytics
- Saved Worksheets - saved analytics worksheets
- Datasets - manage analytics statistics

Concepts

Analytics

Analytics is an advanced facility to graph a variety of statistics in real-time and record this data for later viewing. It has been designed for both long term monitoring and short term analysis. When needed, it makes use of DTrace to dynamically create custom statistics, which allows different layers of the operating system stack to be analyzed in detail.

The following topics provide an overview of how Analytics operates, and links to sections with more details.

Drilldown Analysis

Analytics has been designed around an effective performance analysis technique called *drill-down analysis*. This involves checking high level statistics first, and to focus on finer details based on findings so far. This quickly narrows the focus to the most likely areas.

For example, a performance issue may be experienced and the following high level statistics are checked first:

- Network bytes/sec
- NFSv3 operations/sec
- Disk operations/sec
- CPU utilization

Network bytes/sec is found to be at normal levels, and the same for disk operations and CPU utilization. NFSv3 operations/sec is somewhat high, and the type of NFS operation is then checked and found to be of type "read". So far we have drilled down to a statistic which could be named "NFS operations/sec of type read", which we know is higher than usual.

Some systems may have exhausted available statistics at this point, however Analytics can drill down much further. "NFSv3 operations/sec of type read" can then be viewed *by* client - which means, rather than examining a single graph - we can now see separate graphs for each NFS client. (These separate graphs sum to the original statistic that we had.)

Let's say we find that the host "kiowa" is responsible for a majority of the NFS reads. We can use Analytics to drill down further, to see what files this client is reading. Our statistic becomes "NFSv3 operations/sec of type read for client kiowa broken down by filename". From this, we can see that kiowa is reading through every file on the NFS server. Armed with this information, we can ask the owner of kiowa to explain.

The above example is possible in Analytics, which can keep drilling down further if needed. To summarize, the statistics we examined were:

- "NFSv3 operations/sec"
- "NFSv3 operations/sec by type"
- "NFSv3 operations/sec of type read by client"
- "NFSv3 operations/sec of type read for client kiowa broken down by filename"

These match the statistic names as created and viewed in Analytics.

Statistics

In Analytics, the user picks statistics of interest to display on custom worksheets. Statistics available from Analytics include:

- Network device bytes by device and direction
- NFS operations by filename, client, share, type, offset, size and latency
- SMB operations by filename, client, share, type, offset, size and latency
- Disk operations by type, disk, offset, size and latency
- CPU utilization by CPU-id, mode and application

See the Open Workshetes view for listing statistics, and the Preferences view for enabling advanced Analytics - which will make many more statistics available. The Statistics page discusses available statistics in more detail.

Datasets

A *dataset* refers to all existing data for a particular statistic. Datasets contain:

- Statistic data cached in memory due to the statistic being opened or archived.
- Archived statistic data on disk.

Datasets can be managed in the Datasets view.

Actions

The following actions may be performed on statistics/datasets:

Action	Description
Open	Begin reading from the statistic (every second) and cache values in memory as a dataset. In Open Worksheets, statistics are opened when they are added to the view, allowing them to be graphed in real-time. The data is kept in memory while the statistic is being viewed.
Close	Closes the statistic view, discarding the in memory cached dataset.
Archive	Sets the statistic to be permanently opened and archived to disk. If the statistic had already been opened, then all cached data in memory is also archived to disk. Archiving statistics creates permanent datasets, visible in the Datasets view (those with a non-zero "on disk" value). This is how statistics may be recorded 24x7, so that activity from days, weeks and months in the past can be viewed after the fact.
Destroy	Close the statistic, destroy the dataset and delete all archived data from disk.
Suspend	Pause an archived statistic. New data will not be read, but the existing disk archive will be left intact.
Resume	Resumes a previously suspended statistic, so that it will continue reading data and writing to the archive.

Worksheets

A worksheet is the BUI screen on which statistics are graphed. Multiple statistics can be plotted at the same time, and worksheets may be assigned a title and saved for future viewing. The act of saving a worksheet will automatically execute the archive action on all open statistics - meaning whatever statistics were open, will continue to be read and archived forever.

See the Open Worksheets section for how to drive worksheets, and the Saved Worksheets section for managing previously saved worksheets.

Statistics

Introduction

Analytics statistics provide incredible appliance observability, showing how the appliance is behaving and how clients on the network are using it.

Descriptions

While the statistics presented by Analytics may appear straight forward, there may be additional details to be aware of when interpreting their meaning. This is especially true for the purposes of performance analysis, where precise understanding of the statistics is often necessary. The following pages document each of the available statistics and breakdowns:

Analytics

- CPU: Percent utilization *
- Cache: ARC accesses *
- Cache: L2ARC I/O bytes
- Cache: L2ARC accesses
- Data Movement: NDMP bytes transferred to/from disk
- Data Movement: NDMP bytes transferred to/from tape
- Data Movement: Shadow migration bytes
- Data Movement: Shadow migration ops
- Data Movement: Shadow migration requests
- Disk: Disks *
- Disk: I/O bytes *
- Disk: I/O operations *
- Network: Device bytes
- Network: Interface bytes
- Protocol: SMB operations
- Protocol: Fibre Channel bytes
- Protocol: Fibre Channel operations
- Protocol: FTP bytes
- Protocol: HTTP/WebDAV requests
- Protocol: iSCSI bytes
- Protocol: iSCSI operations
- Protocol: NFSv2 operations
- Protocol: NFSv3 operations
- Protocol: NFSv4 operations
- Protocol: SFTP bytes
- Protocol: SRP bytes

- Protocol: SRP operations

** recommended reading*

Advanced Analytics

These statistics are only visible if Advanced Analytics is enabled in Preferences. These are statistics of lesser interest and are not typically needed for system observability. They are often dynamic which can induce higher overhead, and expose more complex areas of the system which require additional expertise to understand properly:

- CPU: CPUs
- CPU: Kernel spins
- Cache: ARC adaptive parameter
- Cache: ARC evicted bytes
- Cache: ARC size
- Cache: ARC target size
- Cache: DNLC accesses
- Cache: DNLC entries
- Cache: L2ARC errors
- Cache: L2ARC size
- Data Movement: NDMP file system operations
- Data Movement: NDMP jobs
- Disk: Percent utilization
- Disk: ZFS DMU operations
- Disk: ZFS logical I/O bytes
- Disk: ZFS logical I/O operations
- Memory: Dynamic memory usage
- Memory: Kernel memory
- Memory: Kernel memory in use
- Memory: Kernel memory lost to fragmentation
- Network: IP bytes
- Network: IP packets
- Network: TCP bytes
- Network: TCP packets
- System: NSCD backend requests
- System: NSCD operations

Default Statistics

For reference, the following are the statistics that are enabled and archived by default on a factory installed appliance. These are the thirty or so statistics you see in the Datasets view when you first configure and login to the appliance:

Category	Statistic
CPU	percent utilization
CPU	percent utilization broken down by CPU mode
Cache	ARC accesses per second broken down by hit/miss
Cache	ARC size
Cache	ARC size broken down by component
Cache	DNLC accesses per second broken down by hit/miss
Cache	L2ARC accesses per second broken down by hit/miss
Cache	L2ARC size
Data Movement	NDMP bytes transferred to/from disk per second
Disk	Disks with utilization of at least 95 percent broken down by disk
Disk	I/O bytes per second
Disk	I/O bytes per second broken down by type of operation
Disk	I/O operations per second
Disk	I/O operations per second broken down by disk
Disk	I/O operations per second broken down by type of operation
Network	device bytes per second
Network	device bytes per second broken down by device
Network	device bytes per second broken down by direction
Protocol	SMB operations per second
Protocol	SMB operations per second broken down by type of operation
Protocol	FTP bytes per second
Protocol	Fibre Channel bytes per second
Protocol	Fibre Channel operations per second
Protocol	HTTP/WebDAV requests per second
Protocol	NFSv2 operations per second
Protocol	NFSv2 operations per second broken down by type of operation
Protocol	NFSv3 operations per second

Category	Statistic
Protocol	NFSv3 operations per second broken down by type of operation
Protocol	NFSv4 operations per second
Protocol	NFSv4 operations per second broken down by type of operation
Protocol	SFTP bytes per second
Protocol	iSCSI operations per second
Protocol	iSCSI bytes per second

These have been chosen to give broad observability across protocols with minimal statistic collection overhead, and are usually left enabled even when benchmarking. For more discussion on statistic overhead, see Overhead.

Tasks

Statistics Tasks

▼ Determining the impact of a dynamic statistic

For this example task we will determine the impact of "Protocol: NFSv3 operations per second broken down by file name":

1. **Go to Open Worksheets.**
2. **Add the statistic: "Protocol: NFSv3 operations per second as a raw statistic". This is a static statistic and will have negligible performance impact.**
3. **Create steady NFSv3 load; or wait for a period of steady load.**
4. **Add the statistic: "Protocol: NFSv3 operations per second broken down by filename". As this statistic is being created, you may see a temporary sharp dip in performance.**
5. **Wait at least 60 seconds.**
6. **Close the by-filename statistic by clicking on the close icon.**
7. **Wait another 60 seconds.**

8 **Now examine the "Protocol: NFSv3 operations per second as a raw statistic" graph by pausing and zooming out to cover the previous few minutes. Was there a drop in performance when the by-filename statistic was enabled? If the graph looks erratic, try this process again - or try this with a workload that is more steady.**

9 **Click on the graph to see the values at various points, and calculate the percentage impact of that statistic.**

CPU Percent utilization

CPU: Percent Utilization

This shows the average utilization of the appliance CPUs. A CPU may be a core on a socket or a hardware thread; the number and type can be seen under Hardware. For example, a system may have four sockets of quad-core CPUs, meaning there are 16 CPUs available to the appliance. The utilization shown by this statistic is the average across all CPUs.

The appliance CPUs can reach 100% utilization, which may or may not be a problem. For some performance tests the appliance is delibrately driven to 100% CPU utilization to measure it at peak performance.

Example

This example shows CPU: Percent utilization broken down by CPU mode, while the appliance served over 2 Gbytes/sec of cached data over NFSv3:

image

An average of 82% utilization suggests that there could be more headroom available, and that appliance may be able to serve more than 2 Gbytes/sec (it can). (The breakdowns only add to 81%; the extra 1% is due to rounding.)

The high level of CPU utilization does mean that overall latency of NFS operations may increase, which can be measured by Protocol: NFSv3 operations broken down by latency, as operations may be waiting for CPU resources more often.

When to check

When searching for system bottlenecks. This may also be checked when enabling features that consume CPU, such as compression, to gauge the CPU cost of that feature.

Breakdowns

Available breakdowns of CPU Percent utilization:

Breakdown	Description
CPU mode	Either user or kernel. See the CPU modes table below.
CPU identifier	Numeric operating system identifier of the CPU.
application name	Name of the application which is on-CPU.
process identifier	Operating system process ID (PID).
user name	Name of the user who owns the process or thread which is consuming CPU.

The CPU modes are:

CPU mode	Description
user	This is a user-land process. The most common user-land process consuming CPU is akd, the appliance kit daemon, which provides administrative control of the appliance.
kernel	This is a kernel-based thread which is consuming CPU. Many of the appliance services are kernel-based, such as NFS and SMB.

Further Analysis

A problem with this CPU utilization average is that it can hide issues when a single CPU is at 100% utilization, which may happen if a single software thread is saturated with work. Use the Advanced Analytic CPU: CPUs broken down by percent utilization, which represents utilization as a heat map of CPUs, allowing a single CPU at 100% to be easily identified.

Details

CPU utilization represents the time spent processing CPU instructions in user and kernel code, that are not part of the idle thread. Instruction time includes stall cycles on the memory bus, so high utilization can be caused by the I/O movement of data.

Cache ARC accesses

Cache: ARC accesses

The ARC is the Adaptive Replacement Cache, and is an in-DRAM cache for filesystem and volume data. This statistic shows accesses to the ARC, and allows its usage and performance to be observed.

When to check

When investigating performance issues, to check how well the current workload is caching in the ARC.

Breakdowns

Available breakdowns of Cache ARC accesses are:

Breakdown	Description
hit/miss	The result of the ARC lookup. hit/miss states are described in the table below.
file name	The file name that was requested from the ARC. Using this breakdown allows hierarchy mode to be used, so that filesystem directories can be navigated.
L2ARC eligibility	This is the eligibility of L2ARC caching, as measured at the time of ARC access. A high level of ARC misses which are L2ARC eligible would suggest that the workload would benefit from 2nd level cache devices.
project	This shows the project which is accessing the ARC.
share	This shows the share which is accessing the ARC.

As described in Overhead, breakdown such as by file name would be the most expensive to leave enabled.

The hit/miss states are:

hit/miss breakdown	Description
data hits	A data block was in the ARC DRAM cache and returned.
data misses	A data block was not in the ARC DRAM cache. It will be read from the L2ARC cac devices (if available and the data is cached on them) or the pool disks.
metadata hits	A metadata block was in the ARC DRAM cache and returned. Metadata includes t on-disk filesystem framework which refers to the data blocks. Other examples are listed below.
metadata misses	A metadata block was not in the ARC DRAM cache. It will be read from the L2ARC cache devices (if available and the data is cached on them) or the pool disks.
prefetched data/metadata hits/misses	ARC accesses triggered by the prefetch mechanism, not directly from an applicatio request. More details on prefetch follow.

Details

Metadata

Examples of metadata:

- Filesystem block pointers
- Directory information
- Data deduplication tables
- ZFS uberblock

Prefetch

Prefetch is a mechanism to improve the performance of streaming read workloads. It examines I/O activity to identify sequential reads, and can issue extra reads ahead of time so that the data can be in cache before the application requests it. Prefetch occurs *before the ARC* by performing accesses to the ARC - bear this in mind when trying to understand prefetch ARC activity. For example, if you see:

Type	Description
prefetched data missess	prefetch identified a sequential workload, and requested that the data be cached in the ARC ahead of time by performing ARC accesses for that data. The data was not in the cache already, and so this is a "miss" and the data is read from disk. This is normal, and is how prefetch populates the ARC from disk.
prefeteched data hits	prefetch identified a sequential workload, and requested that the data be cached in the ARC ahead of time by performing ARC accesses for that data. As it turned out, the data was already in the ARC - so these accesses returned as "hits" (and so the prefetch ARC access wasn't actually needed). This happens if cached data is repeatedly read in a sequential manner.

After data has been prefetched, the application may then request it with its own ARC accesses. Note that the sizes may be different: prefetch may occur with a 128 Kbyte I/O size, while the application may be reading with an 8 Kbyte I/O size. For example, the following doesn't appear directly related:

- data hits: 368
- prefetch data misses: 23

However it may be: if prefetch was requesting with a 128 KByte I/O size, 23 x 128 = 2944 Kbytes. And if the application was requesting with an 8 Kbyte I/O size, 368 x 8 = 2944 Kbytes.

Further Analysis

To investigate ARC misses, check that the ARC has grown to use available DRAM using Cache: ARC size.

Cache L2ARC IO bytes

Cache: L2ARC I/O bytes

The L2ARC is the 2nd Level Adaptive Replacement Cache, and is an SSD based cache that is accessed before reading from the much slower pool disks. The L2ARC is currently intended for random read workloads. This statistic shows the read and write byte rates to the L2ARC cache devices, if cache devices are present.

When to check

This can be useful to check during warmup. The write bytes will show the rate of L2ARC warmup of time.

Breakdowns

Breakdown	Description
type of operation	read or write. Read bytes are hits on the cache devices. Write bytes show the cache devices populating with data.

Further Analysis

Also see Cache: L2ARC accesses.

Cache L2ARC accesses

Cache: L2ARC accesses

The L2ARC is the 2nd Level Adaptive Replacement Cache, and is an SSD based cache that is accessed before reading from the much slower pool disks. The L2ARC is currently intended for random read workloads. This statistic shows L2ARC accesses if L2ARC cache devices are present, allowing its usage and performance to be observed.

When to check

When investigating performance issues, to check how well the current workload is caching in the L2ARC.

Breakdowns

Breakdown	Description
hit/miss	The result of the L2ARC lookup. hit/miss states are described in the table below.
file name	The file name that was requested from the L2ARC. Using this breakdown allows hierarchy mode to be used, so that filesystem directories can be navigated.
L2ARC eligibility	This is the eligibility of L2ARC caching, as measured at the time of L2ARC access.
project	This shows the project which is accessing the L2ARC.
share	This shows the share which is accessing the L2ARC.

As described in Overhead, breakdown such as by file name would be the most expensive to leave enabled.

Further Analysis

To investigate L2ARC misses, check that the L2ARC has grown enough in size using the Advanced Analytic Cache: L2ARC size. The L2ARC typically takes hours, if not days, to warm up hundreds of Gbytes when feeding from small random reads. The rate can also be checked by examining writes from Cache: L2ARC I/O bytes. Also check the Advanced Analytic Cache: L2ARC errors to see if there are any errors preventing the L2ARC from warming up.

Cache: ARC accesses by L2ARC eligibility can also be checked to see if the data is eligible for L2ARC caching in the first place. Since the L2ARC is intended for random read workloads, it will ignore sequential or streaming read workloads, allowing them to be returned from the pool disks instead.

Data Movement NDMP bytes transferred tofrom disk

Data Movement: NDMP bytes transferred to/from disk

This statistic shows total NDMP bytes transferred per second to or from the local pool disks. It will indicate how much data is being read or written for NDMP backups. This statistic will be zero unless NDMP is configured and active.

When to check

When investigating NDMP backup performance. This can also be checked when trying to identify an unknown disk load, some of which may be caused by NDMP.

Breakdowns

Breakdown	Description
type of operation	read or write.

Further Analysis

Also see Data Movement: NDMP bytes transferred to/from tape.

Data Movement NDMP bytes transferred tofrom tape

Data Movement: NDMP bytes transferred to/from tape

This statistic shows total NDMP bytes per second transferred to or from attached tape devices. This statistic will be zero unless NDMP is configured and active.

When to check

When investigating NDMP backup performance.

Breakdowns

Breakdown	Description
type of operation	read or write.

Further Analysis

Also see Data Movement: NDMP bytes transferred to/from disk.

Data Movement Shadow migration bytes

Data Movement: Shadow migration bytes

This statistic tracks total Shadow Migration bytes per second 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.

When to check

When investigating Shadow Migration activity.

Breakdowns

Breakdown	Description
file name	The file name that was migrated. Using this breakdown allows hierarchy mode to be used, so that filesystem directories can be navigated.
project	This shows the project which contains a shadow migration.
share	This shows the share which is being migrated.

Further Analysis

Also see Data Movement: Shadow migration ops and Data Movement: Shadow migration requests.

Data Movement Shadow migration ops

Data Movement: Shadow migration ops

This statistic tracks Shadow Migration operations that require going to the source filesystem.

When to check

When investigating Shadow Migration activity.

Breakdowns

Breakdown	Description
file name	The file name that was migrated. Using this breakdown allows hierarchy mode to be used, so that filesystem directories can be navigated.
project	This shows the project which contains a shadow migration.

Breakdown	Description
share	This shows the share which is being migrated.
latency	Measure the latency of requests from the shadow migration source.

Further Analysis

Also see Data Movement: Shadow migration bytes and Data Movement: Shadow migration requests.

Data Movement Shadow migration requests

Data Movement: Shadow migration requests

This statistic tracks Shadow Migration 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 can be used to track the latency incurred as part of shadow migration, as well as track the progress of background migration. It currently encompasses both synchronous and asynchronous (background) migration, so it's not possible to view only latency visible to clients.

When to check

When investigating Shadow Migration activity.

Breakdowns

Breakdown	Description
file name	The file name that was migrated. Using this breakdown allows hierarchy mode to be used, so that filesystem directories can be navigated.
project	This shows the project which contains a shadow migration.
share	This shows the share which is being migrated.
latency	Measure the latency incurred as part of shadow migration.

Further Analysis

Also see Data Movement: Shadow migration ops and Data Movement: Shadow migration bytes.

Disk Disks

Disk: Disks

The Disks statistic is used to display the heat map for disks broken down by percent utilization. This is the best way to identify when pool disks are under heavy load. It may also identify problem disks that are beginning to perform poorly, before their behavior triggers a fault and automatic removal from the pool.

When to check

Any investigation into disk performance.

Breakdowns

Breakdown	Description
percent utilization	A heat map with utilization on the Y-axis and each level on the Y-axis colored by the number of disks at that utilization: from light (none) to dark (many).

Interpretation

Utilization is a better measure of disk load than IOPS or throughput. Utilization is measured as the time during which that disk was busy performing requests (see Details below). At 100% utilization the disk may not be able to accept more requests, and additional I/O may wait on a queue. This I/O wait time will cause latency to increase and overall performance to decrease.

In practise, disks with a consistant Utilization of 75% or higher are an indication of heavy disk load.

The heat map allows a particular pathology to be easily identified: a single disk misperforming and reaching 100% utilization (a bad disk). Disks can exhibit this symptom before they fail. Once disks fail, they are automatically removed from the pool with a corresponding alert. This particular problem is during the time *before* they fail, when their I/O latency is increasing and slowing down overall appliance performance, but their status is considered healthy - they have yet to identify any error state. This situation will be seen as a feint line at the top of the heat map, showing that a single disk has stayed at 100% utilization for some time.

Suggested interpretation summary:

Observed	Suggested Interpretation
Most disks consistently over 75%	Available disk resources are being exhausted.

Observed	Suggested Interpretation
Single disk at 100% for several seconds	This can indicate a bad disk that is about to fail.

Further Analysis

To understand the effect of busy disks on I/O, see Disk: I/O operations broken down by latency. For understanding the nature of the I/O, such as IOPS, throughput, I/O sizes and offsets, use Disk: I/O operations and Disk: I/O bytes.

Details

This statistic is actually a measure of percent busy, which serves as a reasonable approximation of percent utilization since the appliance manages the disks directly. Technically this isn't a direct measure of disk utilization: at 100% busy, a disk may be able to accept more requests which it serves concurrently by inserting into and reordering its command queue, or serves from its on-disk cache.

Disk IO bytes

Disk: I/O bytes

This statistic shows the back-end throughput to the disks. This is after the appliance has processed logical I/O into physical I/O based on share settings, and after software RAID as configured by Storage.

For example, an 8 Kbyte write over NFSv3 may became a 128 Kbyte write after the record size is applied from the share settings, which may then become a 256 Kbyte write to the disks after mirroring is applied, plus additional bytes for filesystem metadata. On the same mirrored environment, an 8 Kbyte NFSv3 read may become a 128 Kbyte disk read after the record size is applied, however this doesn't get doubled by mirroring (the data only needs to be read from one half.) It can help to monitor throughput at all layers at the same time to examine this behavior, for example by viewing:

- Network: device bytes - data rate on the network (logical)
- Disk: ZFS logical I/O bytes - data rate to the share (logical)
- Disk: I/O bytes - data rate to the disks (physical)

When to check

To understand the nature of back-end disk I/O, after an issue has already been determined based on disk utilization or latency. It is difficult to identify an issue from disk I/O throughput alone: a single disk may be performing well at 50 Mbytes/sec (sequential I/O), yet poorly at 5 Mbytes/sec (random I/O.)

Using the disk breakdown and the hierarchy view can be used to determine if the JBODs are balanced with disk I/O throughput. Note that cache and log devices will usually have a different throughput profile to the pool disks, and can often stand out as the highest throughput disks when examining by-disk throughput.

Breakdowns

Breakdown	Description
type of operation	read or write.
disk	pool or system disk. This breakdown can identify system disk I/O vs pool disk I/O, and I/O to cache and log devices.

Further Analysis

See Disk: Disks broken down by percent utilization for the best measure of disk utilization. Disk: I/O operations can also be used to examine operations/sec instead of bytes/sec.

Disk IO operations

Disk: I/O operations

This statistic shows the back-end I/O to the disks (disk IOPS). This is after the appliance has processed logical I/O into physical I/O based on share settings, and after software RAID as configured by Storage.

For example, 16 sequential 8 Kbyte NFSv3 writes may become a single 128 Kbyte write sometime later after the data has been buffered in the ARC DRAM cache, which may then become multiple disk writes due to RAID - such as two writes to each half of a mirror. It can help to monitor I/O at all layers at the same time to examine this behavior, for example by viewing:

- Protocol: NFSv3 operations - NFSv3 writes (logical)
- Disk: ZFS logical I/O operations - share I/O (logical)
- Disk: I/O operations - I/O to the disks (physical)

This statistic includes a breakdown of disk I/O latency, which is a direct measure of performance for synchronous I/O, and also useful as a measure of the magnitude of back-end disk load. It is difficult to identify issues from disk IOPS alone without considering latency: a single disk may be performing well at 400 IOPS (sequential and small I/O hitting mostly from the disk's on-board DRAM cache), yet poorly at 110 IOPS (random I/O causing head seek and waiting on disk rotation.)

When to check

Whenever disk performance is investigated, using:

- Disk: I/O operations broken down by latency

This is presented as a heat map allowing the pattern of I/O latency to be observed, and outliers to be easily identified (click the outlier elimination button to view more). Disk I/O latency is often related to the performance of the delivered logical I/O, such as with synchronous reads (non-prefetch), and synchronous writes. There are situations where the latency is not directly related to logical I/O performance, such as asynchronous writes being flushed sometime later to disk, and for prefetch reads.

After an issue has already been determined based on disk I/O latency or utilization, the nature of the disk I/O can be investigated using the other breakdowns, which show disk I/O counts (IOPS). There are no useful IOPS limits per-disk that can be discussed, as such a limit depends on the type of IOPS (random or sequential) and I/O size (large or small). Both of these attributes can be observed using the breakdowns:

- Disk: I/O operations broken down by offset
- Disk: I/O operations broken down by size

Using the disk breakdown and the hierarchy view can also be used to determine if the JBODs are balanced with disk IOPS. Note that cache and log devices will usually have a different I/O profile to the pool disks, and can often stand out as the highest IOPS disks when examining by-disk I/O.

Breakdowns

Breakdown	Description
type of operation	read or write.
disk	pool or system disk. This can be useful to identify system disk I/O vs pool disk I/O, and I/O to cache and log devices.
size	a heat map showing the distribution of I/O sizes.
latency	a heat map showing the latency of disk I/O, as measured from when the I/O was requested to the disk to when the disk returned the completion.
offset	a heat map showing the disk location offset of disk I/O. This can be used to identify random or sequential disk IOPS (often best by vertically zooming the heat map to make out details.)

Further Analysis

See Disk: Disks broken down by percent utilization for the best measure of disk utilization. Disk: I/O bytes can also be used to examine bytes/sec instead of operations/sec.

Network Device bytes

Network: Device bytes

This statistic measures network device activity in bytes/sec. Network devices are the physical network ports, and are shown in the Device column of Network. The measured bytes by this stastistic includes all network payload headers (Ethernet, IP, TCP, NFS/SMB/etc.)

When to check

Network bytes can be used a rough measure of appliance load. It should also be checked whenever performance issues are investigated, especially for 1 Gbit/sec interfaces, in case the bottleneck is the network device. The maximum practical throughput for network devices in each direction (in or out) based on speed:

- 1 Gbit/sec Ethernet: ~120 Mbytes/sec device bytes
- 10 Gbit/sec Ethernet: ~1.16 Gbytes/sec device bytes

If a network device shows a higher rate than these, use the direction breakdown to see the inbound and outbound components.

Breakdowns

Breakdown	Description
direction	in or out, relative to the appliance. For example, NFS reads to the appliance would be show as out(bound) network bytes.
device	network device (see Devices in Network).

Further Analysis

Also see Network: Interface bytes for network throughput at the interface level, instead of the device level.

Network Interface bytes

Network: Interface bytes

This statistic measures network interface activity in bytes/sec. Network interfaces are the logical network interfaces, and are shown in the Interface column of Network. The measured bytes by this stastistic includes all network payload headers (Ethernet, IP, TCP, NFS/SMB/etc.)

Example

See Network: Device bytes for an example of a similar statistic with similar breakdowns.

When to check

Network bytes can be used a rough measure of appliance load. This statistic can be used to see the rate of network bytes through different interfaces. To examine network devices that make up an interface, especially to identify if there are balancing problems with LACP aggregations, use the Network Device bytes statistic.

Breakdowns

Breakdown	Description
direction	in or out, relative to the appliance. For example, NFS reads to the appliance would be show as out(bound) network bytes.
interface	network interface (see Interfaces in Network).

Further Analysis

Also see Network: Device bytes for network throughput at the device level, instead of the interface level.

Protocol SMB operations

Protocol: SMB operations

This statistic shows SMB operations/sec (SMB IOPS) requested by clients to the appliance. Various useful breakdowns are available: to show the client, filename and latency of the SMB I/O.

Example

See Protocol: NFSv3 operations for an example of a similar statistic with similar breakdowns.

When to check

SMB operations/sec can be used as an indication of SMB load, and can be viewed on the dashboard.

Use the latency breakdown when investigating SMB performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen,

along with outliers. If the SMB latency is high, drill down further on latency to identify the type of operation and filename for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client and filename breakdowns, and the filename hierarchy view. It's best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on a busy production server.

Breakdowns

Breakdown	Description
type of operation	SMB operation type (read/write/readX/writeX/...)
client	remote hostname or IP address of the SMB client.
filename	filename for the SMB I/O, if known and cached by the appliance. If the filename is not known it is reported as "<unknown>".
share	the share for this SMB I/O.
project	the project for this SMB I/O.
latency	a heat map showing the latency of SMB I/O, as measured from when the SMB request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the SMB request, and to perform any disk I/O.
size	a heat map showing the distribution of SMB I/O sizes.
offset	a heat map showing the file offset of SMB I/O. This can be used to identify random or sequential SMB IOPS. Use the Disk I/O operations statistic to check whether random SMB IOPS maps to random Disk IOPS after the filesystem and RAID configuration has been applied.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: SMB operations per second of type read broken down by latency" (to examine latency for reads only)
- "Protocol: SMB operations per second for file '/export/fs4/10ga' broken down by offset" (to examine file access pattern for a particular file)
- "Protocol: SMB operations per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Network: Device bytes for a measure of network throughput caused by the SMB activity; Cache: ARC accesses broken down by hit/miss to see how well an SMB read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Protocol Fibre Channel bytes

Protocol: Fibre Channel bytes

This statistic shows Fibre Channel bytes/sec requested by initiators to the appliance.

Example

See Protocol: iSCSI bytes for an example of a similar statistic with similar breakdowns.

When to check

Fibre Channel bytes/sec can be used as an indication of FC load, in terms of throughput. For a deeper analysis of FC activity, see Protocol: Fibre Channel operations.

Breakdowns

Breakdown	Description
initiator	Fibre Channel client initiator
target	local SCSI target
project	the project for this FC request.
lun	the LUN for this FC request.

See the SAN page for terminology definitions.

Further Analysis

See Protocol: Fibre Channel operations for numerous other breakdowns on FC operations; also see Cache: ARC accesses broken down by hit/miss to see how well an FC read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol Fibre Channel operations

Protocol: Fibre Channel operations

This statistic shows Fibre Channel operations/sec (FC IOPS) requested by initiators to the appliance. Various useful breakdowns are available: to show the initiator, target, type and latency of the FC I/O.

Example

See Protocol: iSCSI operations for an example of a similar statistic with similar breakdowns.

When to check

Fibre Channel operations/sec can be used as an indication of FC load, and can also be viewed on the dashboard.

Use the latency breakdown when investigating FC performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the FC latency is high, drill down further on latency to identify the client initiator, the type of operation and LUN for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client initiator are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client initiator, lun and command breakdowns.

Breakdowns

Breakdown	Description
initiator	Fibre Channel client initiator
target	local SCSI target
project	the project for this FC request.
lun	the LUN for this FC request.
type of operation	FC operation type. This shows how the SCSI command is transported by the FC protocol, which can give an idea to the nature of the I/O.

Breakdown	Description
command	SCSI command sent by the FC protocol. This can show the real nature of the requested I/O (read/write/sync-cache/...).
latency	a heat map showing the latency of FC I/O, as measured from when the FC request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the FC request, and to perform any disk I/O.
offset	a heat map showing the file offset of FC I/O. This can be used to identify random or sequential FC IOPS. Use the Disk I/O operations statistic to check whether random FC IOPS maps to random Disk IOPS after the LUN and RAID configuration has been applied.
size	a heat map showing the distribution of FC I/O sizes.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: Fibre Channel operations per second of command read broken down by latency" (to examine latency for SCSI reads only)

Further Analysis

See Protocol: Fibre Channel bytes for the throughput of this FC I/O; also see Cache: ARC accesses broken down by hit/miss to see how well an FC read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol FTP bytes

Protocol: FTP bytes

This statistic shows FTP bytes/sec requested by clients to the appliance. Various useful breakdowns are available: to show the client, user and filename of the FTP requests.

Example

FTP

When to check

FTP bytes/sec can be used as an indication of FTP load, and can be viewed on the dashboard.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client, user and filename breakdowns, and the filename hierarchy view. It may be best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on appliances with high rates of FTP activity.

Breakdowns

Breakdown	Description
type of operation	FTP operation type (get/put/...)
user	username of the client
filename	filename for the FTP operation, if known and cached by the appliance. If the filename is not known it is reported as "<unknown>".
share	the share for this FTP request.
project	the project for this FTP request.
client	remote hostname or IP address of the FTP client.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: FTP bytes per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Cache: ARC accesses broken down by hit/miss to see how well an FTP read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Protocol HTTPWebDAV requests

Protocol: HTTP/WebDAV requests

This statistic shows HTTP/WebDAV requests/sec requested by HTTP clients. Various useful breakdowns are available: to show the client, filename and latency of the HTTP request.

When to check

HTTP/WebDAV requests/sec can be used as an indication of HTTP load, and can also be viewed on the dashboard.

Use the latency breakdown when investigating HTTP performance issues, especially to quantify the magnitude of the issue. This measures the latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the HTTP latency is high, drill down further on latency to identify the file, size and response code for the high latency HTTP requests, and, check other statistics for both

CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client initiator are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client, response code and requested filename breakdowns.

Breakdowns

Breakdown	Description
type of operation	HTTP request type (get/post)
response code	HTTP response (200/404/...)
client	client hostname or IP address
filename	filename requested by HTTP
latency	a heat map showing the latency of HTTP requests, as measured from when the HTTP request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the HTTP request, and to perform any disk I/O.
size	a heat map showing the distribution of HTTP request sizes.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: HTTP/WebDAV operations per second of type get broken down by latency" (to examine latency for HTTP GETs only)
- "Protocol: HTTP/WebDAV requests per second for response code '404' broken down by file name (to see which non-existant files were requested)
- "Protocol: HTTP/WebDAV requests per second for client 'deimos.sf.fishpong.com' broken down by file name" (to examine files requested by a particular client)

Further Analysis

See Network: Device bytes for a measure of network throughput caused by HTTP activity; also see Cache: ARC accesses broken down by hit/miss to see how well an HTTP read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol iSCSI bytes

Protocol: iSCSI bytes

This statistic shows iSCSI bytes/sec requested by initiators to the appliance.

When to check

iSCSI bytes/sec can be used as an indication of iSCSI load, in terms of throughput. For a deeper analysis of iSCSI activity, see Protocol: iSCSI operations.

Breakdowns

Breakdown	Description
initiator	iSCSI client initiator
target	local SCSI target
project	the project for this iSCSI request.
lun	the LUN for this iSCSI request.
client	the remote iSCSI client hostname or IP address

See the SAN page for terminology definitions.

Further Analysis

See Protocol: iSCSI operations for numerous other breakdowns on iSCSI operations; also see Cache: ARC accesses broken down by hit/miss to see how well an iSCSI read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol iSCSI operations

Protocol: iSCSI operations

This statistic shows iSCSI operations/sec (iSCSI IOPS) requested by initiators to the appliance. Various useful breakdowns are available: to show the initiator, target, type and latency of the iSCSI I/O.

When to check

iSCSI operations/sec can be used as an indication of iSCSI load, and can also be viewed on the dashboard.

Use the latency breakdown when investigating iSCSI performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the iSCSI latency is high, drill down further on latency to identify the client initiator, the type of operation and LUN for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client initiator are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client initiator, lun and command breakdowns.

Breakdowns

Breakdown	Description
initiator	iSCSI client initiator
target	local SCSI target
project	the project for this iSCSI request.
lun	the LUN for this iSCSI request.
type of operation	iSCSI operation type. This shows how the SCSI command is transported by the iSCSI protocol, which can give an idea to the nature of the I/O.
command	SCSI command sent by the iSCSI protocol. This can show the real nature of the requested I/O (read/write/sync-cache/...).
latency	a heat map showing the latency of iSCSI I/O, as measured from when the iSCSI request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the iSCSI request, and to perform any disk I/O.
offset	a heat map showing the file offset of iSCSI I/O. This can be used to identify random or sequential iSCSI IOPS. Use the Disk I/O operations statistic to check whether random iSCSI IOPS maps to random Disk IOPS after the LUN and RAID configuration has been applied.
size	a heat map showing the distribution of iSCSI I/O sizes.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: iSCSI operations per second of command read broken down by latency" (to examine latency for SCSI reads only)

Further Analysis

See Protocol: iSCSI bytes for the throughput of this iSCSI I/O; also see Cache: ARC accesses broken down by hit/miss to see how well an iSCSI read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol NFSv2 operations

Protocol: NFSv2 operations

This statistic shows NFSv2 operations/sec (NFS IOPS) requested by clients to the appliance. Various useful breakdowns are available: to show the client, filename and latency of the NFS I/O.

Example

See Protocol: NFSv3 operations for an example of a similar statistic with similar breakdowns.

When to check

NFS operations/sec can be used as an indication of NFS load, and can be viewed on the dashboard.

Use the latency breakdown when investigating NFS performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the NFS latency is high, drill down further on latency to identify the type of operation and filename for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client and filename breakdowns, and the filename hierarchy view. It's best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on a busy production server.

Breakdowns

Breakdown	Description
type of operation	NFS operation type (read/write/getattr/setattr/lookup/...)
client	remote hostname or IP address of the NFS client.
filename	filename for the NFS I/O, if known and cached by the appliance. There are some circumstances where the filename is not known, such as after a cluster failover and when clients continue to operate on NFS filehandles without issuing an open to identify the filename; in these situations the filename reported is "<unknown>".
share	the share for this NFS I/O.
project	the project for this NFS I/O.
latency	a heat map showing the latency of NFS I/O, as measured from when the NFS request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the NFS request, and to perform any disk I/O.
size	a heat map showing the distribution of NFS I/O sizes.
offset	a heat map showing the file offset of NFS I/O. This can be used to identify random or sequential NFS IOPS. Use the Disk I/O operations statistic to check whether random NFS IOPS maps to random Disk IOPS after the filesystem and RAID configuration has been applied.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: NFSv2 operations per second of type read broken down by latency" (to examine latency for reads only)
- "Protocol: NFSv2 operations per second for file '/export/fs4/10ga' broken down by offset" (to examine file access pattern for a particular file)
- "Protocol: NFSv2 operations per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Network: Device bytes for a measure of network throughput caused by the NFS activity; Cache: ARC accesses broken down by hit/miss to see how well an NFS read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Protocol NFSv3 operations

Protocol: NFSv3 operations

This statistic shows NFSv3 operations/sec (NFS IOPS) requested by clients to the appliance. Various useful breakdowns are available: to show the client, filename and latency of the NFS I/O.

When to check

NFS operations/sec can be used as an indication of NFS load, and can be viewed on the dashboard.

Use the latency breakdown when investigating NFS performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the NFS latency is high, drill down further on latency to identify the type of operation and filename for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client and filename breakdowns, and the filename hierarchy view. It's best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on a busy production server.

Breakdowns

Breakdown	Description
type of operation	NFS operation type (read/write/getattr/setattr/lookup/...)
client	remote hostname or IP address of the NFS client.
filename	filename for the NFS I/O, if known and cached by the appliance. There are some circumstances where the filename is not known, such as after a cluster failover and when clients continue to operate on NFS filehandles without issuing an open to identify the filename; in these situations the filename reported is "<unknown>".
share	the share for this NFS I/O.
project	the project for this NFS I/O.

Breakdown	Description
latency	a heat map showing the latency of NFS I/O, as measured from when the NFS request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the NFS request, and to perform any disk I/O.
size	a heat map showing the distribution of NFS I/O sizes.
offset	a heat map showing the file offset of NFS I/O. This can be used to identify random or sequential NFS IOPS. Use the Disk I/O operations statistic to check whether random NFS IOPS maps to random Disk IOPS after the filesystem and RAID configuration has been applied.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: NFSv3 operations per second of type read broken down by latency" (to examine latency for reads only)
- "Protocol: NFSv3 operations per second for file '/export/fs4/10ga' broken down by offset" (to examine file access pattern for a particular file)
- "Protocol: NFSv3 operations per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Network: Device bytes for a measure of network throughput caused by the NFS activity; Cache: ARC accesses broken down by hit/miss to see how well an NFS read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Protocol NFSv4 operations

Protocol: NFSv4 operations

This statistic shows NFSv4 operations/sec (NFS IOPS) requested by clients to the appliance. Various useful breakdowns are available: to show the client, filename and latency of the NFS I/O.

Example

See Protocol: NFSv3 operations for an example of a similar statistic with similar breakdowns.

When to check

NFS operations/sec can be used as an indication of NFS load, and can be viewed on the dashboard.

Use the latency breakdown when investigating NFS performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the NFS latency is high, drill down further on latency to identify the type of operation and filename for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client and filename breakdowns, and the filename hierarchy view. It's best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on a busy production server.

Breakdowns

Breakdown	Description
type of operation	NFS operation type (read/write/getattr/setattr/lookup/...)
client	remote hostname or IP address of the NFS client.
filename	filename for the NFS I/O, if known and cached by the appliance. There are some circumstances where the filename is not known, such as after a cluster failover and when clients continue to operate on NFS filehandles without issuing an open to identify the filename; in these situations the filename reported is "<unknown>".
share	the share for this NFS I/O.
project	the project for this NFS I/O.
latency	a heat map showing the latency of NFS I/O, as measured from when the NFS request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the NFS request, and to perform any disk I/O.
size	a heat map showing the distribution of NFS I/O sizes.
offset	a heat map showing the file offset of NFS I/O. This can be used to identify random or sequential NFS IOPS. Use the Disk I/O operations statistic to check whether random NFS IOPS maps to random Disk IOPS after the filesystem and RAID configuration has been applied.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: NFSv4 operations per second of type read broken down by latency" (to examine latency for reads only)

- "Protocol: NFSv4 operations per second for file '/export/fs4/10ga' broken down by offset" (to examine file access pattern for a particular file)
- "Protocol: NFSv4 operations per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Network: Device bytes for a measure of network throughput caused by the NFS activity; Cache: ARC accesses broken down by hit/miss to see how well an NFS read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Protocol SFTP bytes

Protocol: SFTP bytes

This statistic shows SFTP bytes/sec requested by clients to the appliance. Various useful breakdowns are available: to show the client, user and filename of the SFTP requests.

Example

See Protocol: FTP bytes for an example of a similar statistic with similar breakdowns.

When to check

SFTP bytes/sec can be used as an indication of SFTP load, and can be viewed on the dashboard.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client, user and filename breakdowns, and the filename hierarchy view. It may be best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on appliances with high rates of SFTP activity.

Breakdowns

Breakdown	Description
type of operation	SFTP operation type (get/put/...)
user	username of the client
filename	filename for the SFTP operation, if known and cached by the appliance. If the filename is not known it is reported as "<unknown>".

Breakdown	Description
share	the share for this SFTP request.
project	the project for this SFTP request.
client	remote hostname or IP address of the SFTP client.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: SFTP bytes per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Cache: ARC accesses broken down by hit/miss to see how well an SFTP read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Since SFTP uses SSH to encrypt FTP, there will be additional CPU overhead for this protocol. To check overall CPU utilization of the appliance, see CPU: Percent utilization.

Protocol SRP bytes

Protocol: SRP bytes

This statistic shows SRP bytes/sec requested by initiators to the appliance.

Example

See Protocol: iSCSI bytes for an example of a similar statistic with similar breakdowns.

When to check

SRP bytes/sec can be used as an indication of SRP load, in terms of throughput. For a deeper analysis of SRP activity, see Protocol: SRP operations.

Breakdowns

Breakdown	Description
initiator	SRP client initiator
target	local SCSI target

Breakdown	Description
project	the project for this SRP request.
lun	the LUN for this SRP request.

See the SAN page for terminology definitions.

Further Analysis

See Protocol: SRP operations for numerous other breakdowns on SRP operations; also see Cache: ARC accesses broken down by hit/miss to see how well an SRP read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol SRP operations

Protocol: SRP operations

This statistic shows SRP operations/sec (SRP IOPS) requested by initiators to the appliance. Various useful breakdowns are available: to show the initiator, target, type and latency of the SRP I/O.

Example

See Protocol: iSCSI operations for an example of a similar statistic with similar breakdowns.

When to check

SRP operations/sec can be used as an indication of SRP load.

Use the latency breakdown when investigating SRP performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the SRP latency is high, drill down further on latency to identify the client initiator, the type of operation and LUN for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client initiator are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client initiator, lun and command breakdowns.

Breakdowns

Breakdown	Description
initiator	SRP client initiator
target	local SCSI target
project	the project for this SRP request.
lun	the LUN for this SRP request.
type of operation	SRP operation type. This shows how the SCSI command is transported by the SRP protocol, which can give an idea to the nature of the I/O.
command	SCSI command sent by the SRP protocol. This can show the real nature of the requested I/O (read/write/sync-cache/...).
latency	a heat map showing the latency of SRP I/O, as measured from when the SRP request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the SRP request, and to perform any disk I/O.
offset	a heat map showing the file offset of SRP I/O. This can be used to identify random or sequential SRP IOPS. Use the Disk I/O operations statistic to check whether random SRP IOPS maps to random Disk IOPS after the LUN and RAID configuration has been applied.
size	a heat map showing the distribution of SRP I/O sizes.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: SRP operations per second of command read broken down by latency" (to examine latency for SCSI reads only)

Further Analysis

See Protocol: SRP bytes for the throughput of this SRP I/O; also see Cache: ARC accesses broken down by hit/miss to see how well an SRP read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

CPU CPUs

CPU: CPUs

The CPUs statistic is used to display the heat map for CPUs broken down by percent utilization. This is the most accurate way to examine how CPUs are utilized.

When to check

When investigating CPU load, after checking the utilization average from CPU: Percent utilization.

This statistic is particularly useful for identifying if a single CPU is fully utilized, which can happen if a single thread is saturated with load. If the work performed by this thread cannot be offloaded to other threads so that it can be run concurrently across multiple CPUs, then that single CPU can become the bottleneck. This will be seen as a single CPU stuck at 100% utilization for several seconds or more, while the other CPUs are idle.

Breakdowns

Breakdown	Description
percent utilization	A heat map with utilization on the Y-axis and each level on the Y-axis colored by the number of CPU at that utilization: from light (none) to dark (many).

Details

CPU utilization includes the time to process instructions (that are not part of the idle thread); which includes memory stall cycles. CPU utilization can be caused by:

- executing code (including spinning on locks)
- memory load

Since the appliance primarily exists to move data, memory load often dominates. So a system with high CPU utilization may actually be high as it is moving data.

CPU Kernel spins

CPU: Kernel spins

This statistic counts the number of spin cycles on kernel locks, which consume CPU.

An understanding of operating system internals is required to properly interpret this statistic.

When to check

When investigating CPU load, after checking CPU: Percent utilization and CPU: CPUs broken down by percent utilization.

Some degree of kernel spins is normal for processing any workload, due to the nature of multi-threaded programming. Compare the behavior of kernel spins over time, and for different workloads, to develop an expectation for what is normal.

Breakdowns

Breakdown	Description
type of synchronization primitive	type of lock (mutex/...)
CPU identifier	CPU identifier number (0/1/2/3/...)

Cache ARC adaptive parameter

Cache: ARC adaptive parameter

This is arc_p from the ZFS ARC. This shows how the ARC is adapting its MRU and MFU list size depending on the workload.

An understanding of ZFS ARC internals may be required to properly interpret this statistic.

When to check

Rarely; this may be useful for identifying internal behavior of the ARC, however there are other statistics to check before this one.

If there are caching issues on the appliance, check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload. Then, check the Advanced Analytics Cache: ARC size and Cache: ARC evicted bytes for further details on the ARC behavior.

Breakdowns

none.

Cache ARC evicted bytes

Cache: ARC evicted bytes

This statistic shows bytes that were evicted from the ZFS ARC, as part of its usual housekeeping. The breakdown allows L2ARC eligibility to be examined.

An understanding of ZFS ARC internals may be required to properly interpret this statistic.

When to check

This could be checked if you were considering to install cache devices (L2ARC), as this statistic can be broken down by L2ARC state. If L2ARC eligible data was frequently being evicted from the ARC, then the presence of cache devices could improve performance.

This may also be useful to check if you have issues with cache device warmup. The reason may be that your workload is not L2ARC eligible.

If there are ARC caching issues on the appliance, also check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload. Then, check the Advanced Analytics Cache: ARC size for further details on the ARC behavior.

Breakdowns

Breakdown	Description
L2ARC state	shows L2ARC cached or uncached, L2ARC eligible or ineligible.

Cache ARC size

Cache: ARC size

This statistic shows the size of the primary filesystem cache, the DRAM based ZFS ARC.

An understanding of ZFS ARC internals may be required to properly interpret this statistic.

When to check

When examining the effectiveness of the ARC on the current workload. The ARC should automatically increase in size to fill most of available DRAM, when enough data be accessed by the current workload to be placed in the cache. The breakdown allows the contents of the ARC to be identified by type.

This may also be checked when using cache devices (L2ARC) on systems with limited DRAM, as the ARC can become consumed with L2ARC headers.

If there are ARC caching issues on the appliance, also check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload.

Breakdowns

Available breakdowns:

Breakdown	Description
component	type of data in the ARC. See table below

ARC component types:

Component	Description
ARC data	cached contents, including filesystem data and filesystem metadata.
ARC headers	space consumed by metadata of the ARC itself. The ratio of headers to data is relative to the ZFS record size used; a small record size may mean more ARC headers to refer to the same volume.
ARC other	other kernel consumers of the ARC
L2ARC headers	space consumed by tracking buffers stored on L2ARC devices. If the buffer is on the L2ARC and yet still in ARC DRAM, it is considered "ARC headers" instead.

Cache ARC target size

Cache: ARC target size

This is arc_c from the ZFS ARC. This shows how the target size which the ARC is attempting to maintain. For the actual size, see the Advanced Analytic Cache: ARC size.

An understanding of ZFS ARC internals may be required to properly interpret this statistic.

When to check

Rarely; this may be useful for identifying internal behavior of the ARC, however there are other statistics to check before this one.

If there are caching issues on the appliance, check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload. Then, check the Advanced Analytics Cache: ARC size and Cache: ARC evicted bytes for further details on the ARC behavior.

Breakdowns

none.

Cache DNLC accesses

Cache: DNLC accesses

This statistic shows accesses to the DNLC (Directory Name Lookup Cache). The DNLC caches pathname to inode lookups.

An understanding of operating system internals may be required to properly interpret this statistic.

When to check

This may be useful to check if a workload accesses millions of small files, for which the DNLC can help.

If there are generic caching issues on the appliance, first check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload. Then, check the Advanced Analytic Cache: ARC size for the size of the ARC.

Breakdowns

Breakdown	Description
hit/miss	shows counts for hits/misses, allowing the effectiveness of the DNLC to be checked.

Cache DNLC entries

Cache: DNLC entries

This shows the number of entries in the DNLC (Directory Name Lookup Cache). The DNLC caches pathname to inode lookups.

An understanding of operating system internals may be required to properly interpret this statistic.

When to check

This may be useful to check if a workload accesses millions of small files, for which the DNLC can help.

If there are generic caching issues on the appliance, first check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload. Then, check the Advanced Analytic Cache: ARC size for the size of the ARC.

Breakdowns

none.

Cache L2ARC errors

Cache: L2ARC errors

This statistic shows L2ARC error statistics.

When to check

This may be useful to leave enabled when using cache devices, for when troubleshooting L2ARC issues beyond the standard statistics.

Breakdowns

Available breakdowns:

Breakdown	Description
error	L2ARC error type. See table below.

L2ARC error types:

Error	Description
memory abort	The L2ARC choose not to populate for a one second interval due to a shortage of system memory (DRAM) which holds the L2ARC metadata. Continual memory aborts will prevent the L2ARC from warming up.
bad checksum	A read from a cache device failed the ZFS ARC checksum. This may be an indicator that a cache device is beginning to fail.
io error	A cache device returned an error. This may be an indicator that a cache device is beginning to fail.

Cache L2ARC size

Cache: L2ARC size

This shows the size of data stored on the L2ARC cache devices. This is expected to increase in size over a period of hours or days, until the amount of amount of constant L2ARC eligible data is cached, or the cache devices are full.

When to check

When troubleshooting L2ARC warmup. If the size is small, check that the workload applied should be populating the L2ARC using the statistic Cache: ARC evicted bytes broken down by L2ARC state, and use the Protocol breakdowns such as by size and by offset to confirm that the workload is of random I/O. Sequential I/O does not populate the L2ARC. Another statistic to check is Cache: L2ARC errors.

The L2ARC size does shrink, if data that was cached is deleted from the filesystem.

Breakdowns

none.

Data Movement NDMP file system operations

Data Movement: NDMP file system operations

This statistic shows accesses to the NDMP file system operations/sec.

When to check

This could be useful to check when investigating the source of ZFS load. This would be after checking all other sources of file system activity, via the Protocol statistics. Also see the standard Analytics statistic Data Movement: NDMP bytes transferred to/from disk and Data Movement: NDMP bytes transferred to/from tape.

Breakdowns

Breakdown	Description
type of operation	read/write/...

Data Movement NDMP jobs

Data Movement: NDMP jobs

This statistic shows active NDMP job counts.

When to check

When monitoring NDMP progress, and troubleshooting NDMP. Also see the standard Analytics statistic Data Movement: NDMP bytes transferred to/from disk and Data Movement: NDMP bytes transferred to/from tape.

Breakdowns

Breakdown	Description
type of operation	type of job: backup/restore.

Disk Percent utilization

Disk: Percent utilization

This statistic shows average utilization across all disks. The per-disk breakdown shows the utilization that that disk contributed to the total average, not the utilization of that disk.

When to check

This statistic may be useful to trigger an alert based on the average for all disks.

Investigating disk utilization is usually much more effective using the standard Analytics statistic Disk: Disks broken down by percent utilization - which instead of averaging utilization, presents it as a heat map. This allows individual disk utilization to be examined.

Breakdowns

Breakdown	Description
disk	disks, including system and pool disks.

The disk breakdown shows the contribution to the average percent which each disk made.

Notes

A system with 100 disks would never show more than 1 for any disk breakdown, unless that disk was selected and displayed separately as a raw statistic. Such a system would also show 0 percent utilization for disks less than 50% busy, due to rounding. Since this may be a source of confusion, and that there is a better statistic available for most situations (Disk: Disks), this statistic has been placed in the Advanced category.

See Disk: Disks broken down by percent utilization for a different and usually more effective way to display this data.

Disk ZFS DMU operations

Disk: ZFS DMU operations

This statistic shows ZFS DMU (Data Management Unit) operations/sec.

An understanding of ZFS internals is required to properly interpret this statistic.

When to check

Troubleshooting performance issues, after all relevant standard Analytics have been examined.

The DMU object type breakdown can identify if there is excessive DDT (Data Deduplication Table) activtity. See Data Deduplication.

Breakdowns

Breakdown	Description
type of operation	read/write/...
DMU object level	integer
DMU object type	ZFS plain file/ZFS directory/DMU dnode/SPA space map/...

Disk ZFS logical IO bytes

Disk: ZFS logical I/O bytes

This statistic shows logical access to the ZFS file system as bytes/sec. Logical I/O refers to the type of operations as those that are requested to the file system, such as by NFS; as opposed to physical I/O, which are the requests by the file system to the back-end pool disks.

When to check

This could be useful while investigating how I/O is processed between the Protocol layer and pool disks.

Breakdowns

Breakdown	Description
type of operation	read/write/...
pool name	Name of the disk pool.

Disk ZFS logical IO operations

Disk: ZFS logical I/O operations

This statistic shows logical access to the ZFS file system as operations/sec. Logical I/O refers to the type of operations as those that are requested to the file system, such as by NFS; as opposed to physical I/O, which are the requests by the file system to the back-end pool disks.

When to check

This could be useful while investigating how I/O is processed between the Protocol layer and pool disks.

Breakdowns

Breakdown	Description
type of operation	read/write/...

Breakdown	Description
pool name	Name of the disk pool.

Memory Dynamic memory usage

Memory: Dynamic memory usage

This statistic gives a high level view of memory (DRAM) consumers, updated every second.

When to check

This can be used to check that the filesystem cache has grown to consume available memory.

Breakdowns

Available breakdowns:

Breakdown	Description
application name	See table below.

Application names:

Application Name	Description
cache	The ZFS filesystem cache (ARC). This will grow to consume as much of available memory as possible, as it caches frequently accessed data.
kernel	The operating system kernel.
mgmt	The appliance management software.
unused	Unused space.

Memory Kernel memory

Memory: Kernel memory

This statistic shows kernel memory which is allocated, and can be broken down by kernel cache (kmem cache).

An understanding of operating system internals is required to understand this statistic.

When to check

Rarely. If the dashboard were to show kernel memory as a large consumer of available DRAM (in the Usage: Memory section), then this may be used when troubleshooting the cause. Also see Memory: Kernel memory in use and Memory: Kernel memory lost to fragmentation.

Breakdowns

Breakdown	Description
kmem cache	Kernel memory cache name.

Memory Kernel memory in use

Memory: Kernel memory in use

This statistic shows kernel memory which is in use (populated), and can be broken down by kernel cache (kmem cache).

An understanding of operating system internals is required to understand this statistic.

When to check

Rarely. If the dashboard were to show kernel memory as a large consumer of available DRAM (in the Usage: Memory section), then this may be used when troubleshooting the cause. Also see Memory: Kernel memory lost to fragmentation.

Breakdowns

Breakdown	Description
kmem cache	Kernel memory cache name.

Memory Kernel memory lost to fragmentation

Memory: Kernel memory lost to fragmentation

This statistic shows kernel memory which is currently lost to fragmentation, and can be broken down by kernel cache (kmem cache). Such a state can occur when memory is freed (for example, when cached file system data is deleted), and the kernel has yet to recover the memory buffers.

An understanding of operating system internals is required to understand this statistic.

When to check

Rarely. If the dashboard were to show kernel memory as a large consumer of available DRAM (in the Usage: Memory section), then this may be used when troubleshooting the cause. Also see Memory: Kernel memory in use.

Breakdowns

Breakdown	Description
kmem cache	Kernel memory cache name.

Network IP bytes

Network: IP bytes

This statistic shows IP payload bytes/second, excluding the Ethernet/IB and IP headers.

When to check

Rarely. Network throughput monitoring can be achieved using the standard Analytics statistic Network: Device bytes, which is enabled and achived by default. Examining by-client

throughput can usually be achieved through the Protocol statistic (for example, Protocol: iSCSI bytes, which allows other useful breakdowns based on the protocol). This statistic is most useful if the previous two were not appropriate for some reason.

Breakdowns

Breakdown	Description
hostname	remote client, either as a hostname or IP address.
protocol	IP protocol: tcp/udp
direction	relative to the appliance. in/out

Network IP packets

Network: IP packets

This statistic shows IP packets/second.

When to check

Rarely. Since packets usually map to protocol operations, it is often more useful to examine these using the Protocol statistics (for example, Protocol: iSCSI operations, which allows other useful breakdowns based on the protocol).

Breakdowns

Breakdown	Description
hostname	remote client, either as a hostname or IP address.
protocol	IP protocol: tcp/udp
direction	relative to the appliance. in/out

Network TCP bytes

Network: TCP bytes

This statistic shows TCP payload bytes/second, excluding the Ethernet/IB, IP and TCP headers.

When to check

Rarely. Network throughput monitoring can be achieved using the standard Analytics statistic Network: Device bytes, which is enabled and achived by default. Examining by-client throughput can usually be achieved through the Protocol statistic (for example, Protocol: iSCSI bytes, which allows other useful breakdowns based on the protocol). This statistic is most useful if the previous two were not appropriate for some reason.

Breakdowns

Breakdown	Description
client	remote client, either as a hostname or IP address.
local service	TCP port: http/ssh/215(administration)/...
direction	relative to the appliance. in/out

Network TCP packets

Network: TCP packets

This statistic shows TCP packets/second.

When to check

Rarely. Since packets usually map to protocol operations, it is often more useful to examine these using the Protocol statistics (for example, Protocol: iSCSI operations, which allows other useful breakdowns based on the protocol).

Breakdowns

Breakdown	Description
client	remote client, either as a hostname or IP address.
local service	TCP port: http/ssh/215(administration)/...
direction	relative to the appliance. in/out

System NSCD backend requests

System: NSCD backend requests

This statistic shows requests made by NSCD (Name Service Cache Daemon) to back-end sources, such as DNS, NIS, etc.

An understanding of operating system internals may be required to properly interpret this statistic.

When to check

It may be useful to check the latency breakdown if long latencies were experienced on the appliance, especially during administrative logins. The breakdowns for the database name and source will show what the latency is for, and which remote server is responsible.

Breakdowns

Breakdown	Description
type of operation	request type
result	success/fail
database name	NSCD database (DNS/NIS/...)
source	hostname or IP address of this request
latency	time for this request to complete

System NSCD operations

System: NSCD operations

This statistic shows requests made to NSCD (Name Service Cache Daemon).

An understanding of operating system internals may be required to properly interpret this statistic.

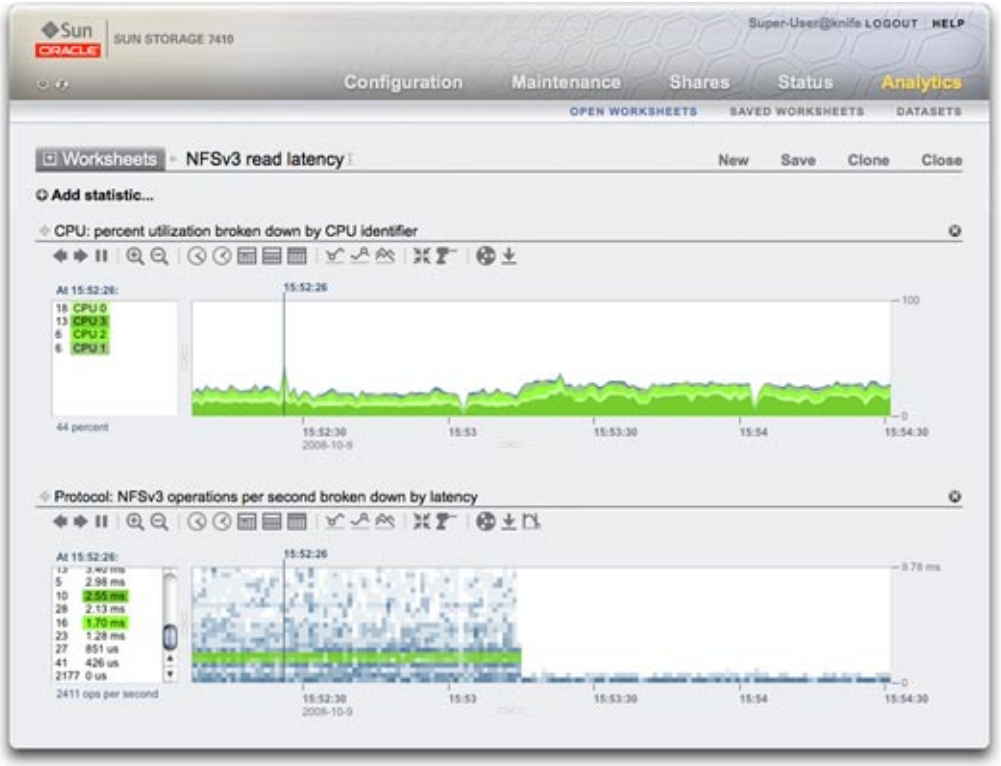
When to check

This can be used to check the effectiveness of the NSCD cache, by using the hit/miss breakdown. Misses become backend requests to remote sources, which can be examined using System: NSCD backend requests.

Breakdowns

Breakdown	Description
type of operation	request type
result	success/fail
database name	NSCD database (DNS/NIS/...)
latency	time for this request to complete
hit/miss	result from the cache lookup: hit/miss

Open Worksheets



Using Analytics to examine CPU utilization and NFSv3 operation latency

Worksheets

This is the main interface for Analytics. See Concepts for an overview of Analytics.

A worksheet is a view where multiple statistics may be graphed. The screenshot at the top of this page shows two statistics:

- CPU: percent utilization broken down by CPU identifier - as a *graph*
- Protocol: NFSv3 operations per second broken down by latency - as a *quantize plot*

Click the screenshot for a larger view. The following sections introduce Analytics features based on that screenshot.

Graph

The CPU utilization statistic in the screenshot is rendered as a graph. Graphs provide the following features:

- The left panel lists components of the graph, if available. Since this graph was "... broken down by CPU identifier", the left panel lists CPU identifiers. Only components which had activity in the visible window (or selected time) will be listed on the left.
- Left panel components can be clicked to highlight their data in the main plot window.
- Left panel components can be shift clicked to highlight multiple components at a time (such as in this example, with all four CPU identifiers highlighted).
- Left panel components can be right clicked to show available drilldowns.
- Only ten left panel components will be shown to begin with, followed by "...". You can click the "..." to reveal more. Keep clicking to expand the list completely.
- The graph window on the right can be clicked to highlight a point in time. In the example screenshot, 15:52:26 was selected. Click the pause button followed by the zoom icon to zoom into the selected time. Click the time text to remove the vertical time bar.
- If a point in time is highlighted, the left panel of components will list details for that point in time only. Note that the text above the left box reads "At 15:52:26:", to indicate what the component details are for. If a time wasn't selected, the text would read "Range average:".
- Y-axis auto scales to keep the highest point in the graph (except for utilization statistics, where are fixed at 100%).
- The line graph button  will change this graph to plot just lines without the flood-fill. This may be useful for a couple of reasons: some of the finer detail in line plots can be lost in the flood fill, and so selecting line graphs can improve resolution. This feature can also be used to vertical zoom into component graphs: first, select one or more components on the left, then switch to the line graph.

Quantize Plot

The NFS latency statistic in the screenshot is rendered as a quantize plot. The name refers to the how the data is collected and displayed. For each statistic update, data is quantized into buckets, which are drawn as blocks on the plot. The more events in that bucket for that second, the darker the block will be drawn.

The example screenshot shows NFSv3 operations were spread out to 9 ms and beyond - with latency on the y-axis - until an event kicked in about half way and the latency dropped to less than 1 ms. Other statistics can be plotted to explain the drop in latency (the filesystem cache hit rate showed steady misses go to zero at this point - a workload had been randomly reading from disk (0 to 9+ ms latency), and switched to reading files that were cached in DRAM.)

Quantize plots are used for I/O latency, I/O offset and I/O size, and provide the following features:

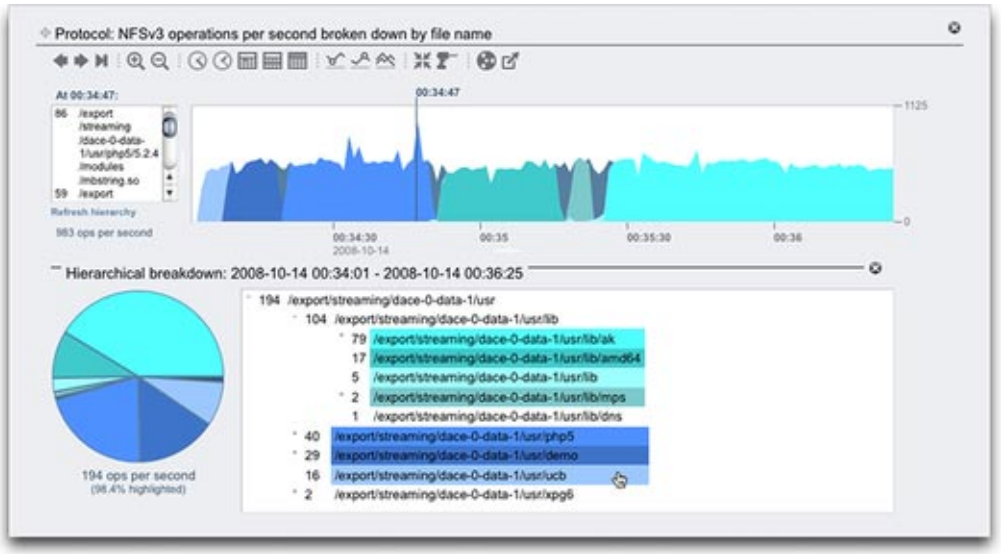
- Detailed understanding of data profile (not just the average, maximum or minimum) these visualize all events and promote pattern identification.

- Vertical outlier elimination. Without this, the y-axis would always be compressed to include the highest event. Click the crop outliers icon to toggle between different percentages of outlier elimination. Mouse over this icon to see the current value.
- Vertical zoom: click a low point from the list in the left box, then shift-click a high point. Now click the crop outliers icon to zoom to this range.

Show Hierarchy

Graphs by filename have a special feature - "Show hierarchy" text will be visible on the left. When clicked, a pie-chart and tree view for the traced filenames will be made available.

The following screenshot shows the hierarchy view:



As with graphs, the left panel will show components based on the statistic break down, which in this example was by filename. Filenames can get a little too long for that left panel - try expanding it by clicking and dragging the divider between it and the graph; or use the hierarchy view.

The hierarchy view provides the following features:

- The filesystem may be browsed, by clicking "+" and "-" next to file and directory names.
- File and directory names can be clicked, and their component will shown in the main graph.
- Shift click pathnames to display multiple components at once, as shown in this screenshot.
- The pie chart on the left shows the ratio of each component to the total.

- Slices of the pie may be clicked to perform highlighting.
- If the graph isn't paused, the data will continue to scroll. The hierarchy view can be refreshed to reflect the data visible in the graph by clicking "Refresh hierarchy".

There is a close button on the right to close the hierarchy view.

Common

The following features are common to graphs and quantize plots:

- The height may be expanded. Look for a white line beneath in the middle of the graph, click and drag downwards.
- The width will expand to match the size of your browser.
- Click and drag the move icon to switch vertical location of the statistics.

Background Patterns

Normally graphs are displayed with various colors against a white background. If data is unavailable for any reason the graph will be filled with a pattern to indicate the specific reason for data unavailability:

- The gray pattern indicates that the given statistic was not being recorded for the time period indicated. This is either because the user had not yet specified the statistic or because data gathering had been explicitly suspended.
- The red pattern indicates that data gathering was unavailable during that period. This is most commonly seen because the system was down during the time period indicated.
- The orange pattern indicates an unexpected failure while gathering the given statistic. This can be caused by a number of aberrant conditions. If it is seen persistently or in critical situations, contact your authorized support resource and/or submit a support bundle.

Saving a Worksheet

Worksheets can be saved for later viewing. As a side effect, all visible statistics will be archived - meaning that they will continue to save new data after the saved worksheet has been closed.

To save a worksheet, click the "Untitled worksheet" text to name it first, then click "Save" from the local navigation bar. Saved worksheets can be opened and managed from the Saved Worksheets section.

Toolbar Reference

A toolbar of buttons is shown above graphed statistics. The following is a reference for their function:

Icon	Click	Shift-Click
	move backwards in time (moves left)	move backwards in time (moves left)
	move forwards in time (moves right)	move forwards in time (moves right)
	forward to now	forward to now
	pause	pause
	zoom out	zoom out
	zoom in	zoom in
	show one minute	show two minutes, three, four, ...
	show one hour	show two hours, three, four, ...
	show one day	show two days, three, four, ...
	show one week	show two weeks, three, four, ...
	show one month	show two months, three, four, ...
	show minimum	show next minimum, next next minimum, ...
	show maximum	show next maximum, next next maximum, ...
	show line graph	show line graph
	show mountain graph	show mountain graph
	crop outliers	crop outliers
	sync worksheet to this statistic	sync worksheet to this statistic
	unsync worksheet statistics	unsync worksheet statistics
	drilldown	rainbow highlight

Icon	Click	Shift-Click
	save statistical data	save statistical data
	export statistical data	export statistical data

Mouse over each button to see a tooltip to describe the click behavior.

CLI

Viewing analytics statistics is possible from the CLI. See:

- Reading Datasets - for listing recent statistics from available datasets.
- Saved Worksheets:CLI - for how to dump worksheets in CSV, which may be suitable for automated scripting.

Tips

- If you'd like to save a worksheet that displays an interesting event, make sure the statistics are paused first (sync all statistics, then hit pause). Otherwise the graphs will continue to scroll, and when you open the worksheet later the event may no longer be on the screen.
- If you are analyzing issues after the fact, you will be restricted to the datasets that were already being archived. Visual correlations can be made between them when the time axis is synchronized. If the same pattern is visible in different statistics - there is a good chance that it is related activity.
- Be patient when zooming out to the month view and longer. Analytics is clever about managing long period data, however there can still be delays when zooming out to long periods.

Tasks

BUI

▼ Monitoring NFSv3 or SMB by operation type

1. **Click the add icon.**
2. **Click the "NFSv3 operations" or "SMB operations" line.**
3. **Click "Broken down by type of operation".**

▼ Monitoring NFSv3 or SMB by latency

1. Click the add icon.
2. Click the "NFSv3 operations" or "SMB operations" line.
3. Click "Broken down by latency".

▼ Monitoring NFSv3 or SMB by filename

1. Click the add icon.
2. Click the "NFSv3 operations" or "SMB operations" line.
3. Click "Broken down by filename".
4. When enough data is visible, click the "Show hierarchy" text on the left to display a pie-chart and tree-view for the path names that were traced in the graph.
5. Click "Refresh hierarchy" when the pie-chart and tree-view become out of date with the scrolling data in the graph.

▼ Saving a worksheet

1. Click the "Untitled worksheet" text and type in a custom name
2. Click "Save" from the local navigation bar.

Saved Worksheets

Introduction

Open Worksheets may be saved for at least these reasons:

- To create custom performance views which display statistics of interest.
- To investigate performance events for later analysis. A worksheet may be paused on a particular event and then saved, so that others can open the worksheet later and study the event.

Properties

The following properties are stored for saved worksheets:

Field	Description
Name	Configurable name of the saved worksheet. This will be displayed at the top of the Open Worksheets view
Comment	Optional comment (only visible in the BUI)
Owner	User who owns the worksheet
Created	Time the worksheet was created
Modified	Time the worksheet was last modified (only visible in the CLI)

BUI

Mouse over saved worksheet entries to expose the following controls:

icon	description
	This will upload a support bundle that includes this worksheet, allowing for off-line analysis of your system by your support provider. You should only do this if you have been explicitly asked to upload such a bundle by support personnel.
	Append the datasets saved in this worksheet to the current worksheet in Open Worksheets
	Edit the worksheet to change the name and comment
	Destroy this worksheet

Single click an entry to open that worksheet. This may take several seconds if the worksheet was paused on a time in the distant past, or if it spanned many days, as the appliance must read the statistic data from disk back into memory.

CLI

Worksheet maintenance actions are available under the `analytics worksheets` context. Use the `show` to view the current saved worksheets:

```
walu:> analytics worksheets
walu:analytics worksheets> show
Worksheets:
```

```
WORKSHEET       OWNER   NAME
worksheet-000   root    830 MB/s NFSv3 disk
worksheet-001   root    8:27 event
```

Worksheets may be selected so that more details may be viewed. Here one of the statistics is dumped and retrieved in CSV format from the saved worksheet:

```
walu:analytics worksheets> select worksheet-000
walu:analytics worksheet-000> show
Properties:
                          uuid = e268333b-c1f0-401b-97e9-ff7f8ee8dc9b
                          name = 830 MB/s NFSv3 disk
                         owner = root
                         ctime = 2009-9-4 20:04:28
                         mtime = 2009-9-4 20:07:24

Datasets:

DATASET      DATE        SECONDS NAME
dataset-000  2009-9-4         60 nic.kilobytes[device]
dataset-001  2009-9-4         60 io.bytes[op]

walu:analytics worksheet-000> select dataset-000 csv
Time (UTC),KB per second
2009-09-04 20:05:38,840377
2009-09-04 20:05:39,890918
2009-09-04 20:05:40,848037
2009-09-04 20:05:41,851416
2009-09-04 20:05:42,870218
2009-09-04 20:05:43,856288
2009-09-04 20:05:44,872292
2009-09-04 20:05:45,758496
2009-09-04 20:05:46,865732
2009-09-04 20:05:47,881704
[...]
```

If there was a need to gather Analytics statistics using an automated CLI script over SSH, it would be possible to create a saved worksheet containing the desired statistics which could then be read in this fashion. This is one way to view analytics from the CLI; also see Reading datasets.

Datasets

Introduction

The term *dataset* refers to the in memory cached and on disk saved data for a statistic, and is presented as an entity in Analytics with administration controls.

Datasets are automatically created when statistics are viewed in Open Worksheets, but are not saved to disk for future viewing unless they are *archived*. See the Actions section of Concepts.

BUI

The Analytics->Datasets page in the BUI lists all datasets. These include open statistics that are being viewed in a worksheet (and as such are temporary datasets - they will disappear when the worksheet is closed), and statistics that are being archived to disk.

The following fields are displayed in the Dataset view for all datasets:

Field	Description
Status icon	See below table
Name	Name of the statistic/dataset
Since	First timestamp in dataset. For open statistics, this is the time the statistic was opened - which may be minutes earlier. For archived statistics, this is the first time in the archived dataset which indicates how far back in the past this dataset goes - which may be days, weeks, months. Sorting this column will show the oldest datasets available.
On Disk	Space this dataset consumes on disk
In Core	Space this dataset consumers in main memory

The following icons are visible in the BUI view; some of these will only be visible during mouse over of a dataset entry:

icon	description
	Dataset is actively collecting data
	Dataset is currently suspended from collecting data
	Suspend/Resume archived datasets
	Enable archiving of this dataset to disk
	Destroy this dataset

See Actions for descriptions for these dataset actions.

CLI

The `analytics datasets` context allows management of datasets.

Viewing available datasets

Use the show command to list datasets:

```
caji:analytics datasets> show
Datasets:

DATASET     STATE   INCORE ONDISK NAME
dataset-000 active    674K  35.7K arc.accesses[hit/miss]
dataset-001 active    227K  31.1K arc.l2_accesses[hit/miss]
dataset-002 active    227K  31.1K arc.l2_size
dataset-003 active    227K  31.1K arc.size
dataset-004 active    806K  35.7K arc.size[component]
dataset-005 active    227K  31.1K cpu.utilization
dataset-006 active    451K  35.6K cpu.utilization[mode]
dataset-007 active   57.7K      0 dnlc.accesses
dataset-008 active    490K  35.6K dnlc.accesses[hit/miss]
dataset-009 active    227K  31.1K http.reqs
dataset-010 active    227K  31.1K io.bytes
dataset-011 active    268K  31.1K io.bytes[op]
dataset-012 active    227K  31.1K io.ops
...
```

Many of the above datasets are archived by default, there is only one that is additional: "dataset-007", which has no ONDISK size, indicating that it is a temporary statistic that isn't archived. The names of the statistics are abbreviated versions of what is visible in the BUI: "dnlc.accesses" is short for "Cache: DNLC accesses per second".

Specific dataset properties can be viewed after selecting it:

```
caji:analytics datasets> select dataset-007
caji:analytics dataset-007> show
Properties:
                        name = dnlc.accesses
                    grouping = Cache
                 explanation = DNLC accesses per second
                      incore = 65.5K
                        size = 0
                   suspended = false
```

Reading datasets

Datasets statistics can be read using the read command, followed by the number of previous seconds to display:

```
caji:analytics datasets> select dataset-007
caji:analytics dataset-007> read 10
DATE/TIME                 /SEC       /SEC BREAKDOWN
2009-10-14 21:25:19        137          - -
2009-10-14 21:25:20        215          - -
2009-10-14 21:25:21        156          - -
2009-10-14 21:25:22        171          - -
2009-10-14 21:25:23       2722          - -
2009-10-14 21:25:24        190          - -
```

```
2009-10-14 21:25:25         156          - -
2009-10-14 21:25:26         166          - -
2009-10-14 21:25:27         118          - -
2009-10-14 21:25:28        1354          - -
```

Breakdowns will also be listed if available. The following shows CPU utilization broken down CPU mode (user/kernel), which was available as `dataset-006`:

```
caji:analytics datasets> select dataset-006
caji:analytics dataset-006> read 5
DATE/TIME                %UTIL      %UTIL BREAKDOWN
2009-10-14 21:30:07          7          6 kernel
                                        0 user
2009-10-14 21:30:08          7          7 kernel
                                        0 user
2009-10-14 21:30:09          0          - -
2009-10-14 21:30:10         15         14 kernel
                                        1 user
2009-10-14 21:30:11         25         24 kernel
                                        1 user
```

The summary is shown in "%UTIL", and contributing elements in "%UTIL BREAKDOWN". At 21:30:10, there 14% kernel time and 1% user time. The 21:30:09 line shows 0% in the "%UTIL" summary, and so does not list breakdowns ("--").

Suspending and Resuming all datasets

The CLI has a feature that is not yet available in the BUI: the ability to suspend and resume all datasets. This may be useful when benchmarking the appliance to determine its absolute maximum performance. Since some statistics can consume significant CPU and disk resources to archive, benchmarks performed with these statistics enabled are invalid.

To suspend all datasets use `suspend`:

```
caji:analytics datasets> suspend
This will suspend all datasets. Are you sure? (Y/N) y
caji:analytics datasets> show
Datasets:

DATASET     STATE   INCORE ONDISK NAME
dataset-000 suspend   638K   584K arc.accesses[hit/miss]
dataset-001 suspend   211K   172K arc.l2_accesses[hit/miss]
dataset-002 suspend   211K   133K arc.l2_size
dataset-003 suspend   211K   133K arc.size
...
```

To resume all datasets use `resume`:

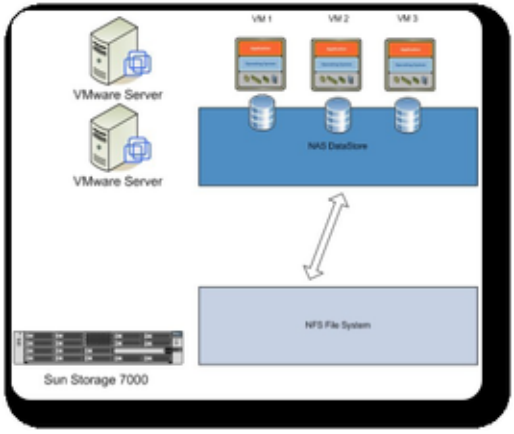
```
caji:analytics datasets> resume
caji:analytics datasets> show
Datasets:

DATASET     STATE   INCORE ONDISK NAME
```

```
dataset-000 active    642K   588K arc.accesses[hit/miss]
dataset-001 active    215K   174K arc.l2_accesses[hit/miss]
dataset-002 active    215K   134K arc.l2_size
dataset-003 active    215K   134K arc.size
...
```

◆ ◆ ◆ CHAPTER 7

Application Integration

Application Integration



Introduction

In some cases, host-side components are required for full application integration with Sun Storage 7000 appliances. The following sections give overviews of these integrations with links and notes for storage administrators. Complete documentation for the application integrations are packaged with the downloaded components.

- Sun Storage 7000 Provider for Microsoft VSS Software
- Sun Storage 7000 Management Plug-In for Oracle Enterprise Manager 10g Grid Controller

Quickly integrate the appliance with existing applications using standard protocols for data. Best practices exist for integrating with applications where special tuning is helpful to get better performance or a more robust environment, for example:

URL	Title
http://www.sun.com/bigadmin/features/articles/storage_vmware.pdf (http://www.sun.com/bigadmin/features/articles/storage_vmware.pdf)	Using Sun Storage 7000 Systems With VMware ESX Server
http://www.sun.com/bigadmin/features/articles/7000_oracle_rac_2009q3.pdf (http://www.sun.com/bigadmin/features/articles/7000_oracle_rac_2009q3.pdf)	Deploying Oracle Real Application Clusters 10g or 11g Release 1 on Sun Storage 7000 Unified Storage Systems (2009.Q3)
http://www.sun.com/bigadmin/features/articles/7000_oracle_deploy_2009q3.pdf (http://www.sun.com/bigadmin/features/articles/7000_oracle_deploy_2009q3.pdf)	Deploying Oracle Databases Using NFS on Sun Storage 7000 Unified Storage Systems (2009.Q3)
http://www.sun.com/bigadmin/features/articles/7000_oracle_iscsi_asm_2009q3.pdf (http://www.sun.com/bigadmin/features/articles/7000_oracle_iscsi_asm_2009q3.pdf)	Deploying Oracle 10g or 11gR1 ASM Using iSCSI on Sunâ¢ Storage 7000 Unified Storage Systems (2009.Q3)

Microsoft

Sun Storage 7000 Provider for Microsoft VSS Software

The Sun Storage 7000 provider for Microsoft VSS Software allows the appliance to take consistent snapshots from Windows hosts which are using block targets. VSS coordinates snapshots to be sure block data is consistent. The service communicates with a set of workflows

to coordinate taking of snapshots as seen from the application, not just with respect to the ZFS file system running on the appliance. It works over iSCSI, FC and SRP.

Volume Shadow Copy Services (VSS) for Microsoft operating systems provides a coordination point between backup infrastructure from a variety of applications that make use of iSCSI LUNs and used by a host system. 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 Sun Storage 7000 provider for VSS is installed on hosts that require this functionality and coordination between applications. Complete documentation for the application integrations are packaged with the downloaded components. The provider, release notes, and information on certifications with third-party applications is available at the Sun Download Center]. More information on VSS is available on the Microsoft web site, including this [http://msdn.microsoft.com/en-us/library/aa384649 (http://msdn.microsoft.com/en-us/library/aa384649)%28VS.85%29.aspx overview.

It is highly recommended that a user with limited access privileges be used from each host leveraging VSS capabilities in conjunction with the appliance. This allows the audit capabilities of the appliance to be used, as well as providing an additional layer of security to ensure other clients are prevented from using credentials reserved for VSS operations. To create limited-access login credentials for use with VSS:

- Create a *vss* role with the following authorizations:
- *nas.*.*.* changeAccessProps, changeGeneralProps, changeProtocolProps, changeSpaceProps, changeUserQuota, clone, createShare, destroy, rename, rollback, scheduleSnap, takeSnap, createProject : workflow.*.* read
- Create a *vss* user and assign it the role *vss*. Communicate this user's login and password to system administrators that intend to install the VSS provider on a supported Windows platform.

Oracle

Sun Storage 7000 Management Plug-In for Oracle Enterprise Manager 10g Grid Controller

The Sun Storage 7000 plug-in for Oracle Enterprise Manager 10g grid controller provides first-class monitoring to the grid controller environment for the Sun Storage 7000 appliance family with the ability to:

- Monitor Sun Storage 7000 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 URL:
https://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_SMI-Site/en_US/-/USD/ViewProductDet

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.

Oracle Sun Storage 7000 Management Plug-In for Oracle Enterprise Manager 10g Grid Controller

Configure 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 will be created with limited access to the system, to allow the Oracle Enterprise Manager 10g 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 Sun Storage 7000 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.

Unconfigure 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, and
- Removes the *Oracle Enterprise Manager* worksheet.

This workflow will *not* disable Advanced Analytics or any of the datasets that were activated for collection purposes.

Glossary

7110	Sun Storage 7110 Unified Storage System
7120	Sun ZFS Storage 7120
7210	Sun Storage 7210 Unified Storage System
7310	Sun Storage 7310 Unified Storage System
7320	Sun ZFS Storage 7320
7410	Sun Storage 7410 Unified Storage System
7420	Sun ZFS Storage 7420
7720	Sun ZFS Storage 7720
Active Directory	Microsoft Active Directory server
Alerts	Configurable log, email or SNMP trap events
Analytics	appliance feature for graphing real-time and historic performance statistics
ARC	Adaptive Replacement Cache
BUI	Browser User Interface
CLI	Command Line Interface
Cluster	Multiple heads connected to shared storage
Controller	See "Storage Controller"
CPU	Central Prcessing Unit
CRU	Customer Replaceable Component
Dashboard	appliance summary display of system health and activity
Dataset	the in-memory and on-disk data for a statistic from Analytics
DIMM	dual in-line memory module
Disk Shelf	the expansion storage shelf that is connected to the head node or storage controller
DNS	Domain Name Service

DTrace	a comprehensive dynamic tracing framework for troubleshooting kernel and application problems on production systems in real-time
FC	Fibre Channel
FRU	Field Replaceable Component
FTP	File Transfer Protocol
GigE	Gigabit Ethernet
HBA	Host Bus Adapter
HCA	Host Channel Adapter
HDD	Hard Disk Drive
HTTP	HyperText Transfer Protocol
Hybrid Storage Pool	combines disk, flash, and DRAM into a single coherent and seamless data store.
Icons	icons visible in the BUI
iSCSI	Internet Small Computer System Interface
Kiosk	a restricted BUI mode where a user may only view one specific screen
L2ARC	Level 2 Adaptive Replacement Cache
LDAP	Lightweight Directory Access Protocol
LED	light-emitting diode
Logzilla	write IOPS accelerator
LUN	Logical Unit
Masthead	top section of BUI screen
Modal Dialog	a new screen element for a specific function
NFS	Network File System
NIC	Network Interface Card
NIS	Network Information Service
PCIe	Peripheral Component Interconnect Express
Pool	provide storage space that is shared across all filesystems and LUNs
Project	a collection of shares
PSU	Power Supply Unit

QDR	quad data rate
Readzilla	read-optimized flash SSD for the L2ARC
Remote Replication	replicating shares to another appliance
Rollback	reverts all of the system software and all of the metadata settings of the system back to their state just prior to applying an update
SAS	Serial Attached SCSI
SAS-2	Serial Attached SCSI 2.0
SATA	Serial ATA
Schema	configurable properties for shares
Scripting	automating CLI tasks
Service	appliance service software
Share	ZFS filesystem shared using data protocols
SIM	SAS Interface Module
Snapshot	an image of a share
SSD	Solid State Drive
SSH	Secure Shell
Statistic	a metric visible from Analytics
Storage Controller	the head node of the appliance
Support Bundle	auto-generated files containing system configuration information and core files for use by remote support in debugging system failures
Title Bar	local navigation and function section of BUI screen
Updates	software or firmware updates
WebDAV	Web based Distributed Authoring and Versioning
ZFS	on-disk data storage subsystem

Index

A

Active Directory, 221, 222
  Joining a Domain, 225
  Joining a Workgroup, 225
Alerts, 135, 138
  Adding a threshold alert, 139
  Adding an alert action, 138–139

D

Dashboard, 52, 54, 55, 57, 58
  Running the Dashboard Continuously, 58
Dataset, 352, 414, 415, 416
DNS, 233, 234, 235

F

FC, 97, 98, 99, 100, 101
FTP, Allowing FTP access to a share, 204

H

HTTP, 204, 205, 376
  Allowing HTTP access to a share, 206

I

Identity Mapping
  Adding a Name-Based Mapping, 232
Identity Mapping *(Continued)*
  Configuring Identity Mapping, 232
iSCSI, 109, 110, 111, 112

L

L2ARC, 361, 362
LDAP, 218, 219, 221
  Adding an appliance administrator from LDAP, 221

M

Masthead, 27

N

Network
  Adding a static route, 85–86, 86
  Create an Infiniband partition datalink and interface, 85
  Create an IPMP group using link-state only failure detection, 84
  Create an IPMP group using probe-based and link-state failure detection, 84
  Creating a single port interface, 82–83
  Creating a single port interface, drag-and-drop, 83
  Creating an LACP aggregated link interface, 83–84
  Deleting a static route, 86
  Extend an IPMP group, 85
  Extend an LACP aggregation, 85

Network *(Continued)*
Modifying an interface, 83
NFS, Sharing a filesystem over NFS, 187
NIS, 217, 218
Adding an appliance administrator from NIS, 218
NTP, BUI Clock Synchronization, 239

O
Open Worksheets
Monitoring NFSv3 or SMB by filename, 412
Monitoring NFSv3 or SMB by latency, 412
Monitoring NFSv3 or SMB by operation type, 411
Saving a worksheet, 412

P
Pool, 87, 90
Project, 311

R
Remote Replication, 323, 327, 338

S
Settings
Changing the Activity Thresholds, 61
Changing the Displayed Activity Statistics, 60
SFTP, Allowing SFTP access to a share, 214
Share, 277, 278, 282
SMB
Active Directory Configuration, 201
Initial Configuration, 199–201
Project and Share Configuration, 201
SMB Data Service Configuration, 201–202
Snapshot, 305, 306, 308, 309, 320
SNMP
Configuring SNMP to send traps, 245
Configuring SNMP to serve appliance status, 245
SSH, Disabling root SSH access, 249
Statistics, Determining the impact of a dynamic statistic, 356–357
Storage, Configuring a Storage Pool, 91

U
Users
Adding a role, 130, 132
Adding a user who can only view the dashboard, 133
Adding an administrator, 130, 131
Adding authorizations to a role, 131, 132
Deleting authorizations from a role, 131, 132

V
Virus Scan, Configuring virus scanning for a share, 216–217

W
WebDAV, 376