Oracle® ZFS Storage Appliance Administration Guide

Part No: E52872-01
June 2014

ORACLE®

Contents

Using This Documentation

- **Overview** – Describes how to administer the Oracle ZFS Storage Appliance
- **Audience** – Technicians, system administrators, and authorized service providers
- **Required knowledge** – Experience working with the Oracle ZFS Storage Appliance

Product Documentation Library

Product Documentation Library Visit `http://www.oracle.com/goto/ZFSStorage/docs` for the Oracle ZFS Storage Appliance documentation library.

For related documentation, including white papers, visit `http://www.oracle.com/technetwork/server-storage/sun-unified-storage/overview/index.html` and click on the Documentation tab. For late-breaking information and known issues about this product, visit My Oracle Support at `http://support.oracle.com`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Feedback

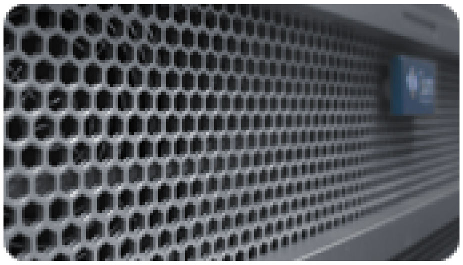
Provide feedback about this documentation at `http://www.oracle.com/goto/docfeedback`.

♦ ♦ ♦ CHAPTER 1

Oracle ZFS Storage Appliance Overview

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

ZFSSA Key Features



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

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

Supported Protocols

The ZFSSA supports a variety of industry-standard client protocols, including the following:

- "SMB" on page 202
- "NFS" on page 195
- "HTTP and HTTPS" on page 219
- "WebDAV" on page 219
- "iSCSI" on page 200
- "SAN Fibre Channel" on page 109
- "SRP" on page 127
- "iSER Target Configuration" on page 122
- "FTP" on page 217
- "SFTP" on page 229

ZFSSA Data Services

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

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

- RAID-Z (RAID-5 and RAID-6), mirrored, and striped Chapter 5, "Storage Configuration"
- Unlimited read-only and read-write "Shares - Snapshots" on page 326, with snapshot schedules
- "Data deduplication" on page 304
- Built-in "data compression" on page 304
- Chapter 13, "Replication" of data for disaster recovery
- Active-active Chapter 10, "Cluster Configuration" for high availability
- Thin provisioning of "iSCSI" on page 200 "LUNs"
- "Virus scanning and quarantine" on page 233
- "NDMP backup and restore" on page 221

Data Availability

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

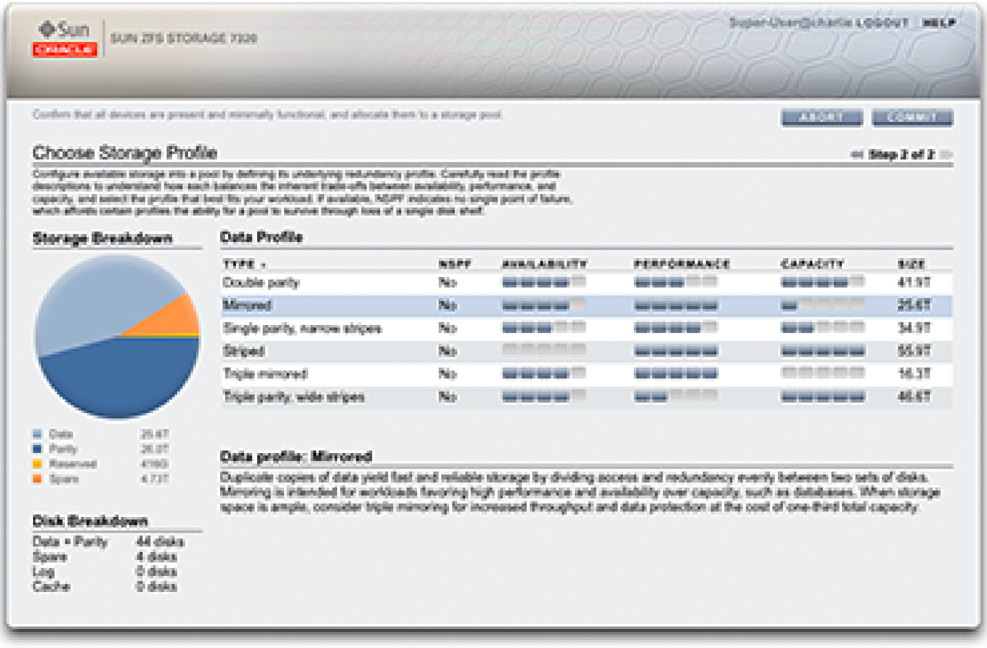
- Predictive self-healing and diagnosis of all system hardware failures: CPUs, DRAM, I/O cards, disks, fans, power supplies
- ZFS end-to-end data checksums of all data and metadata, protecting data throughout the stack
- RAID-6 (double- and triple-parity) and optional RAID-6 across disk shelves
- Active-active Chapter 10, “Cluster Configuration” for high availability
- Chapter 4, “Network Configuration” for network failure protection
- I/O Multipathing between the controller and disk shelves
- Integrated software restart of all system Chapter 11, “ZFSSA Services”
- “Phone-Home” on page 261 of telemetry for all software and hardware issues
- Lights-out Management of each system for remote power control and console access

ZFSSA Configuration

To configure the ZFSSA, use the following sections:

- Chapter 3, “Initial Configuration” - initial configuration
- Chapter 4, “Network Configuration” - networking
- Chapter 11, “ZFSSA Services” - data services
- Chapter 6, “Storage Area Network Configuration” - storage area network configuration
- Chapter 10, “Cluster Configuration” - clustering
- Chapter 7, “User Configuration” - user accounts and access control
- Chapter 7, “User Configuration” - user preferences
- Chapter 9, “Alert Configuration” - custom alerts
- Chapter 5, “Storage Configuration” - reconfigure storage devices
- Chapter 12, “Shares, Projects, and Schema”

Browser User Interface (BUI)



Confirm that all devices are present and minimally functional, and allocate them to a storage pool.

Choose Storage Profile

Configure available storage into a pool by defining its underlying redundancy profile. Carefully read the profile descriptions to understand how each balances the inherent trade-offs between availability, performance, and capacity, and select the profile that best fits your workload. If available, NSPF indicates no single point of failure, which affords certain profiles the ability for a pool to survive through loss of a single disk shelf.

Storage Breakdown

■	Data	25.6T
■	Parity	26.0T
■	Reserved	416G
■	Spare	4.73T

Disk Breakdown

Data + Parity	44 disks
Spare	4 disks
Log	0 disks
Cache	0 disks

Data Profile

TYPE	NSPF	AVAILABILITY	PERFORMANCE	CAPACITY	SIZE
Double parity	No	■■■■■□	■■■■□□	■■■■■□	41.5T
Mirrored	No	■■■■■□	■■■■■■	■□□□□□	25.6T
Single parity, narrow stripes	No	■■■■□□	■■■■■□	■■□□□□	34.5T
Striped	No	□□□□□□	■■■■■■	■■■■■■	55.5T
Triple mirrored	No	■■■■■□	■■■■■■	□□□□□□	16.3T
Triple parity, wide stripes	No	■■■■■□	■■□□□□	■■■■■■	46.6T

Data profile: Mirrored

Duplicate copies of data yield fast and reliable storage by dividing access and redundancy evenly between two sets of disks. Mirroring is intended for workloads favoring high performance and availability over capacity, such as databases. When storage space is ample, consider triple mirroring for increased throughput and data protection at the cost of one-third total capacity.

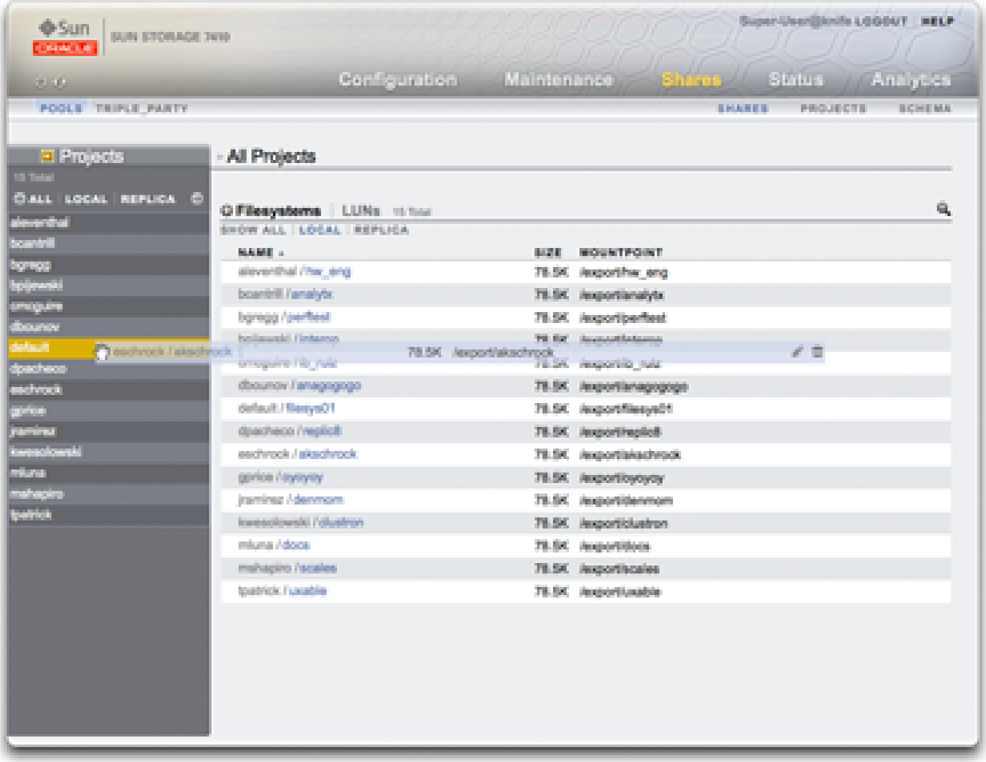
The ZFSSA Browser User Interface (BUI) is the graphical tool for administration of the appliance. The BUI provides an intuitive environment for administration tasks, visualizing concepts, and analyzing performance data. The BUI provides an uncluttered environment for visualizing system behavior and identifying performance issues with the appliance.

Direct your browser to the system using either the *IP address* or *host name* you assigned to the NET-0 port during initial configuration as follows: https://ipaddress:215 or https://hostname:215. The login screen appears.

The online help linked in the top right of the BUI is context-sensitive. For every top-level and second-level screen in the BUI, the associated help page appears when you click the Help button.

- “Main Window” on page 25 - overview of BUI elements and design
- “General Usage” on page 30 - icon reference
- “Supported Browsers” on page 35 - supported browsers

Main Window

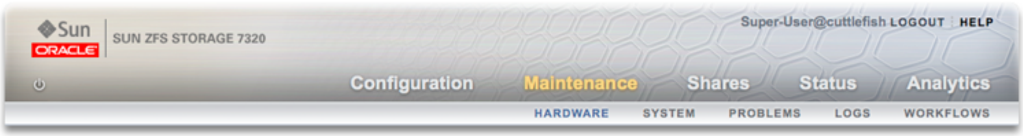


NAME	SIZE	MOUNTPOINT
aleventhal / hw_eng	78.5K	/export/hw_eng
bcantrill / analytx	78.5K	/export/analytx
bgregg / perftest	78.5K	/export/perftest
bpijewski / interop	78.5K	/export/interop
eschrock / akschrock	78.5K	/export/akschrock
cmcguire / lb_ruiz	78.5K	/export/lb_ruiz
dbounov / anagogogo	78.5K	/export/anagogogo
default / filesys01	78.5K	/export/filesys01
dpacheco / replic8	78.5K	/export/replic8
eschrock / akschrock	78.5K	/export/akschrock
gprice / oyoyoy	78.5K	/export/oyoyoy
jramirez / denmom	78.5K	/export/denmom
kwesolowski / clustron	78.5K	/export/clustron
mluna / docs	78.5K	/export/docs
mshapiro / scales	78.5K	/export/scales
tpatrick / usable	78.5K	/export/usable

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

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/Oracle logo, a hardware model badge, and hardware power off/restart button. Across the right, again from top to bottom: login identification, logout, help, main navigation, and subnavigation.



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" on page 48 screen or the full log available on the "Logs" in "Oracle ZFS Storage Appliance Customer Service Manual " screen.

Navigation

Use main navigation links to view between the Chapter 4, "Network Configuration", "Maintenance" in "Oracle ZFS Storage Appliance Customer Service Manual ", Chapter 12, "Shares, Projects, and Schema", Chapter 2, "Status", and "Analytics" in "Oracle ZFS Storage Appliance Analytics Guide " areas of the BUI.

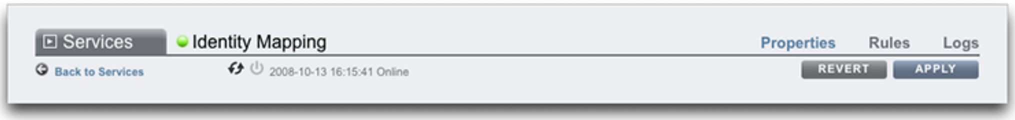
Use sub-navigation links to access features and functions within each area.

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 Chapter 7, "User Configuration" 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.



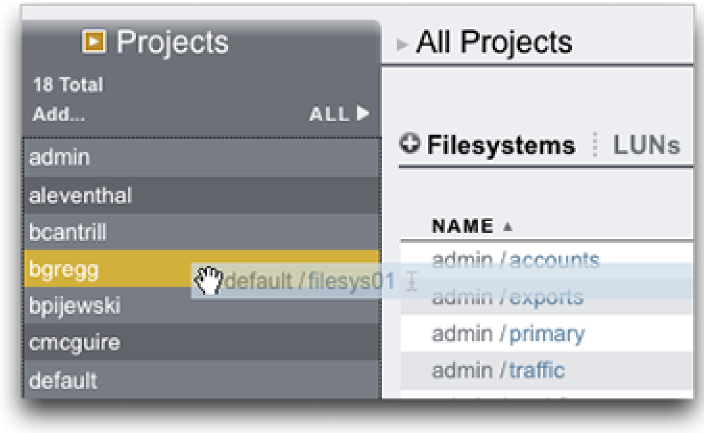
For example, the Identity mapping service title bar enables the following:

- 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

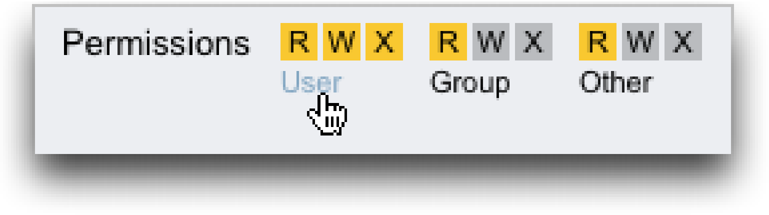
Most BUI controls use standard web form inputs, however there are a few key exceptions worth noting:

TABLE 1-1 Key Web Form Exceptions

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 model
Automatically open side panel	Drag an item to the side panel

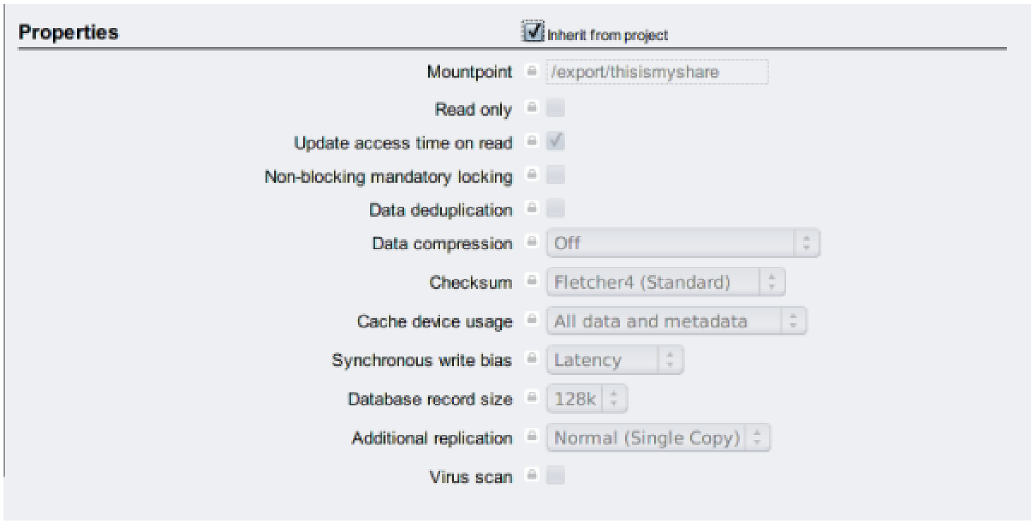
Permissions

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.

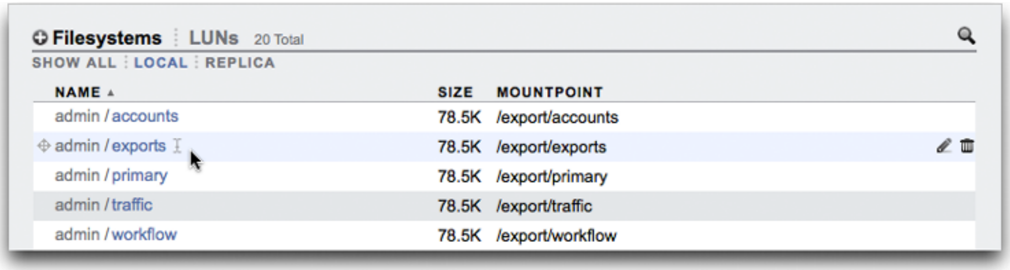


Properties ☑ Inherit from project

Mountpoint	/export/thisismyshare
Read only	□
Update access time on read	☑
Non-blocking mandatory locking	□
Data deduplication	□
Data compression	Off
Checksum	Fletcher4 (Standard)
Cache device usage	All data and metadata
Synchronous write bias	Latency
Database record size	128k
Additional replication	Normal (Single Copy)
Virus scan	□

Viewing List Item Controls

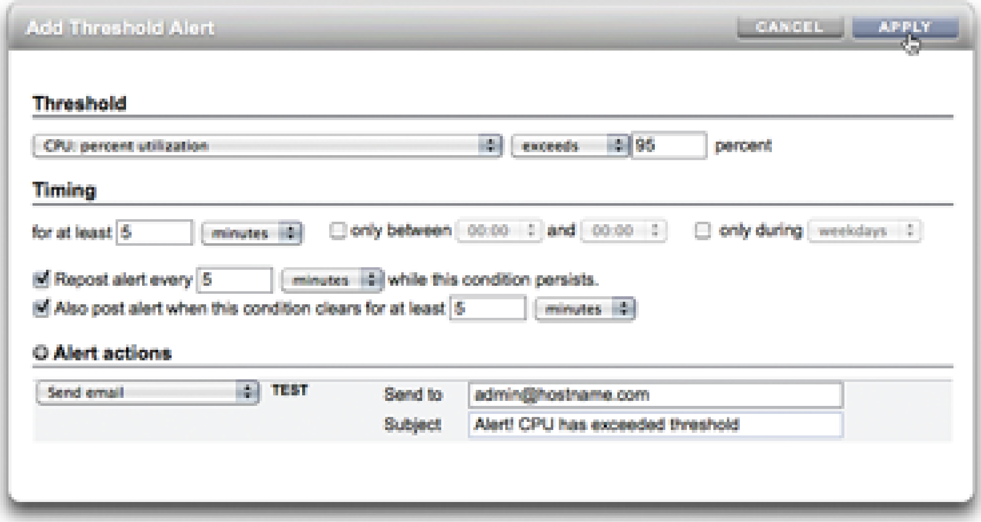
To view controls for an item in a list, hover the mouse over the row.



NAME ▲	SIZE	MOUNTPOINT
admin / accounts	78.5K	/export/accounts
admin / exports	78.5K	/export/exports
admin / primary	78.5K	/export/primary
admin / traffic	78.5K	/export/traffic
admin / workflow	78.5K	/export/workflow

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.



CPU: percent utilization exceeds 95 percent

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

☑ Repost alert every 5 minutes while this condition persists.

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

Send email

Send to admin@hostname.com

Subject Alert! CPU has exceeded threshold

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:

TABLE 1-2 Status Indicators

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:

TABLE 1-3 BUI Icons

Icon*		Description	Icon*		Description
--	⌶	rename (edit text)	--	⇤	sever
--	✥	move	--	⊞	clone
✎	✎	edit	--	↺	rollback
🗑	🗑	destroy	--	⏻	appliance power
⊕	⊕	add	--	➔	apply
⊖	⊖	remove	--	⬅	revert
⊗	⊗	cancel/close	--	ℹ	info
--	!	error	--	▼	sort list column (down)
--	⚠	alert	--	▲	sort list column (up)
⏻	⏻	on/off toggle	⏮	⏮	first page
⟳	⟳	restart	⏪	⏪	previous page

Icon*		Description	Icon*		Description
--	☼	locate	⏩	⏩	next page
⊘	⊘	disable/offline	⏭	⏭	last page
🔒	🔒	lock	--	🔍	search
--	⁘	wait spinner	▼	▼	menu
--	↻	reverse direction	▶	▶	panel

** Disabled icons are shown at left.*

Networking

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

TABLE 1-4 Network Icons

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” on page 57.

TABLE 1-5 Dashboard Icons

Icon	Description	Icon	Description
☀	sunny	🌀	hurricane
⛅	partly cloudy	🌀2	hurricane class 2
☁	cloudy	🌀3	hurricane class 3
💧	rainy	🌀4	hurricane class 4
⚡	stormy	🌀5	hurricane class 5

Analytics

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

TABLE 1-6 Analytics Toolbar Icons

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

Icon	Description	Icon	Description
🕒	show one hour	🔧	drilldown
▦	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.

TABLE 1-7 Identity Mapping Icons

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.

TABLE 1-8 Miscellaneous Icons

Icon	Description	Icon	Description
○	allow	⊡	SAS

Icon	Description	Icon	Description
)(	deny	▬	SAS port
⬡	storage pool		

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 or later
- Internet Explorer 7 or later
- Safari 3.1 or later
- Google Chrome (Stable)
- 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.

Command Line Interface (CLI)

The CLI is designed to mirror the capabilities of the BUI, while also providing a powerful scripting environment for performing repetitive tasks. The command line is an efficient and powerful tool for repetitive administrative tasks. The appliance presents a CLI available through either the “Console” in “Oracle ZFS Storage Appliance Installation Guide ”, or “SSH” on page 276. There are several situations in which the preferred interaction with the system is the CLI, as follows:

- Network unavailability - If the network is unavailable, browser-based management is impossible; the only vector for management is the “Console” in “Oracle ZFS Storage Appliance Installation Guide ”, 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
- 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.
-

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
```

Logging Into the CLI

To log in remotely using the CLI, use an `ssh` client. If you have not Chapter 7, “User Configuration” 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:>
```

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 Chapter 4, “Network 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 Chapter 6, “Storage Area Network Configuration” view in the browser, including Chapter 4, “Network Configuration”, Chapter 11, “ZFSSA Services” and Chapter 7, “User Configuration”, “Preferences”Chapter 8, “Setting ZFSSA 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
John Doe                 bmc                      12345      Dir
Super-User               root                     0          Loc
```

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

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

Last Context

Use the `last` command to navigate to a previously selected or created context. This command is presently implemented in only the replication action context.

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

```
dory:shares p1/share replication> list
           TARGET          STATUS     NEXT
action-000  oakmeal         idle       Sync now
action-001  dory            idle       Sync now
dory:shares p1/share replication> create
dory:shares p1/share action (uncommitted)> set target=dory
                       target = dory (uncommitted)
dory:shares p1/share action (uncommitted)> set pool=p0
                         pool = p0 (uncommitted)
dory:shares p1/share action (uncommitted)> commit
dory:shares p1/share replication> last
dory:shares p1/share action-002> get id
                           id = 7034367a-d4d8-e26f-fa93-c3b454e3b595
dory:shares p1/share action-002> done
dory:shares p1/share replication> select action-000
dory:shares p1/share action-000> get id
                           id = 9895d9f4-7b23-ebe1-faf2-d85a581e3dff
dory:shares p1/share action-000> done
dory:shares p1/share replication> last get id
                           id = 9895d9f4-7b23-ebe1-faf2-d85a581e3dff
dory:shares p1/share replication>
```

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 Chapter 9, “Alert Configuration” 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" on page 200 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" on page 258'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". For information, see Chapter 12, "Shares, Projects, and Schema".

```
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. For information, see Chapter 12, “Shares, Projects, and Schema”.

```
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"
```

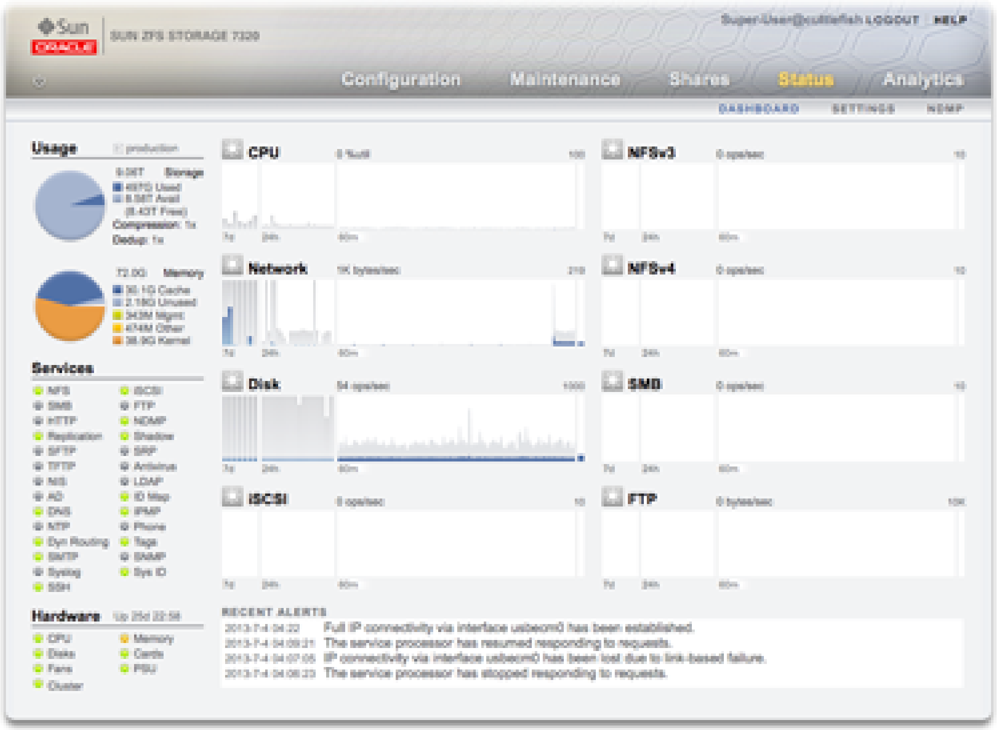
♦ ♦ ♦ CHAPTER 2

Status

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

- The "Status > Dashboard" on page 48 screen provides a view of storage, memory, services, hardware, activity, and recent alerts.
- The "Status > Settings" on page 57 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" on page 60 screen provides a view of any configured NDMP devices and recent activity for each NDMP session.

Dashboard



RECENT ALERTS

2013-7-4 04:22 Full IP connectivity via interface usbecm0 has been established.
2013-7-4 04:09:21 The service processor has resumed responding to requests.
2013-7-4 04:07:05 IP connectivity via interface usbecm0 has been lost due to link-based failure.
2013-7-4 04:06:23 The service processor has stopped responding to requests.

The Dashboard summarizes appliance status

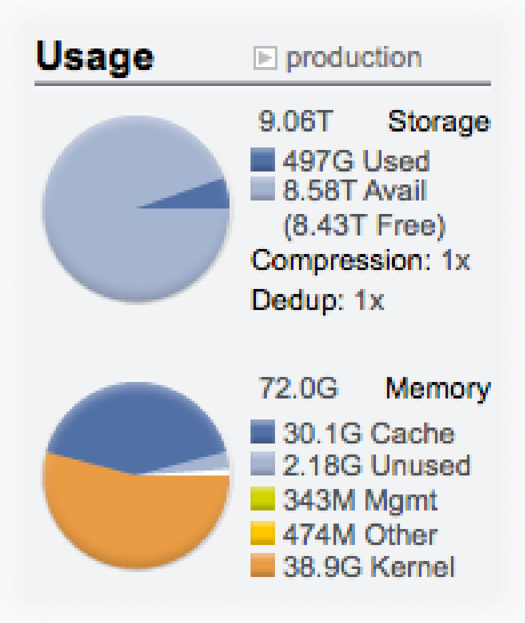
Links

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 border 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 name of the pool appears at the top right of the Usage area. If multiple pools are configured, use the pull-down list to select the desired pool to display.

FIGURE 2-1 Status Dashboard Usage



Storage

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

Memory

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

TABLE 2-1 Summary of Pool Usage

Summary Pool Usage	
Used	Space used by this pool including data and snapshots.

Summary Pool Usage	
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.
Free	Amount of space available, within the LUN capacity, less unused space that is reserved by projects and shares within a pool. Provides the free disk space available when disk space is allocated by reservation in advance and/or when LUNs are created.
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.

TABLE 2-2 Summary of Main Memory Usage

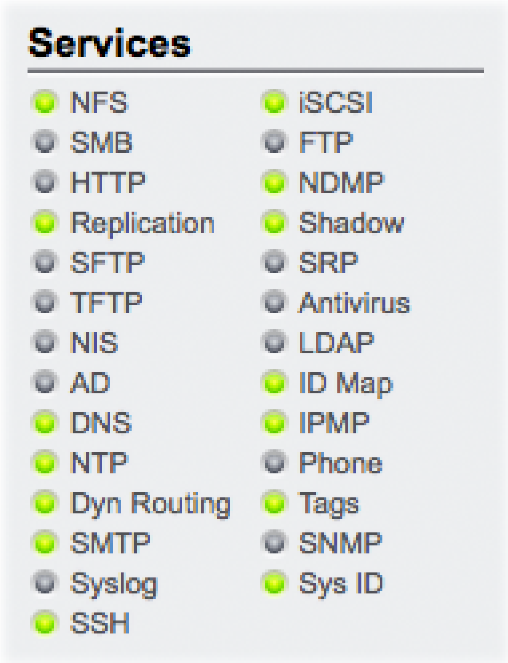
Summary of main memory (RAM) usage	
Cache	Bytes in use by the filesystem cache to improve performance.
Unused	Bytes not currently in use. After booting, this value will decrease as space is used by the filesystem cache.
Mgmt	Bytes in use by the appliance management software.
Other	Bytes in use by miscellaneous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

FIGURE 2-2 Services Dashboard



Services

🟢 NFS
⚪ SMB
⚪ HTTP
🟢 Replication
⚪ SFTP
⚪ TFTP
⚪ NIS
⚪ AD
🟢 DNS
🟢 NTP
🟢 Dyn Routing
🟢 SMTP
⚪ Syslog
🟢 SSH
🟢 iSCSI
⚪ FTP
🟢 NDMP
🟢 Shadow
⚪ SRP
⚪ Antivirus
⚪ LDAP
🟢 ID Map
🟢 IPMP
⚪ Phone
🟢 Tags
⚪ SNMP
🟢 Sys ID

Icons

Most services are green to indicate that the service is online, or grey to indicate that the service is disabled. See the "General Usage" on page 30 section for a reference of all possible states and icon colors.

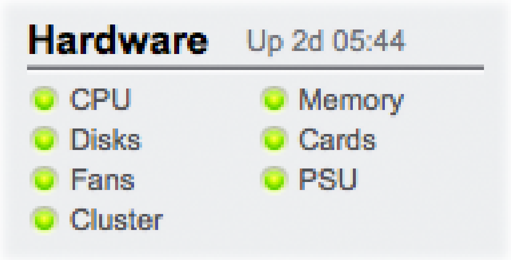
Links

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.

FIGURE 2-3 Hardware Dashboard



Faults

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

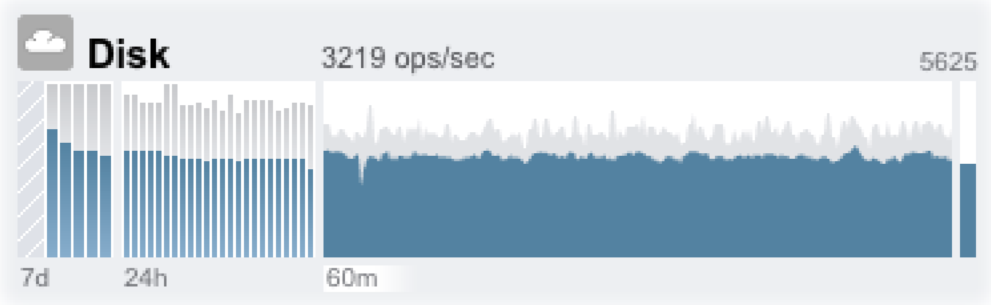
Links

To go to the "Hardware" in "Oracle ZFS Storage Appliance Customer Service Manual " 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.

FIGURE 2-4 Disk Activity Dashboard



To go to the “Analytics” in “Oracle ZFS Storage Appliance Analytics Guide ” 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” on page 57 screen.

Graphs

TABLE 2-3 Summary of Statistic 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 percent.

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

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

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

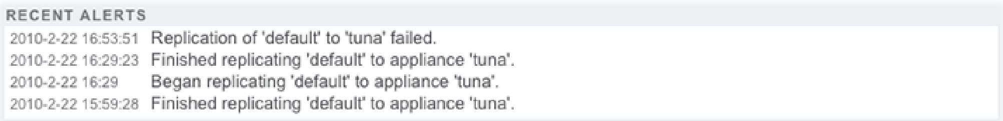
Weather

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" in "Oracle ZFS Storage Appliance Analytics Guide " worksheets for an accurate understanding of system performance.

Recent Alerts

FIGURE 2-5 Recent Alerts



RECENT ALERTS

```
2010-2-22 16:53:51  Replication of 'default' to 'tuna' failed.
2010-2-22 16:29:23  Finished replicating 'default' to appliance 'tuna'.
2010-2-22 16:29     Began replicating 'default' to appliance 'tuna'.
2010-2-22 15:59:28  Finished replicating 'default' to appliance 'tuna'.
```

This section shows the last four appliance alerts. Click the box to go to the “Logs” in “Oracle ZFS Storage Appliance Customer Service Manual ” 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`:

```
cuttlefish:> status dashboard
Storage:
   pool_0:
      Used     497G bytes
      Avail    8.58T bytes
      Free     8.43T bytes
      State          online
      Compression    1x

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

Services:
   ad                disabled               smb              disabled
   dns               online                 ftp               disabled
   http              online                 identity          online
   idmap             online                 ipmp              online
   iscsi             online                 ldap              disabled
   ndmp              online                 nfs               online
   nis               online                 ntp               online
   routing           online                 scrk              maintenance
   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:
   2013-6-15 07:46: A cluster interconnect link has been restored.
```

The previous descriptions in the “BUI” on page 48 section apply, with the following differences:

- The activity plots are not 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:>
```

Running the Dashboard Continuously

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

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

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

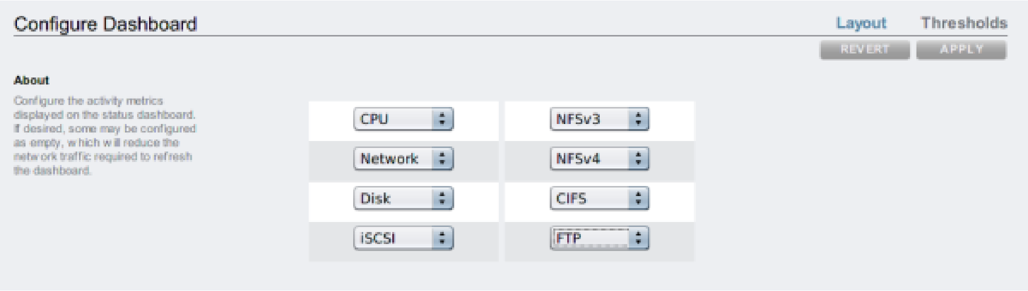
Settings

Introduction

The Status > Settings screen enables you to customize the “Status Dashboard” on page 48, including the statistics that appear and thresholds that indicate activity through the weather icons.

BUI

FIGURE 2-6 Dashboard Settings



Configure the activity metrics displayed on the status dashboard. If desired, some may be configured as empty, which will reduce the network traffic required to refresh the dashboard.

Layout

Use the layout tab to select the graphs that appear in the “dashboard activity” on page 48 area, as defined in the following table.

TABLE 2-4 Status Layout Settings

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.

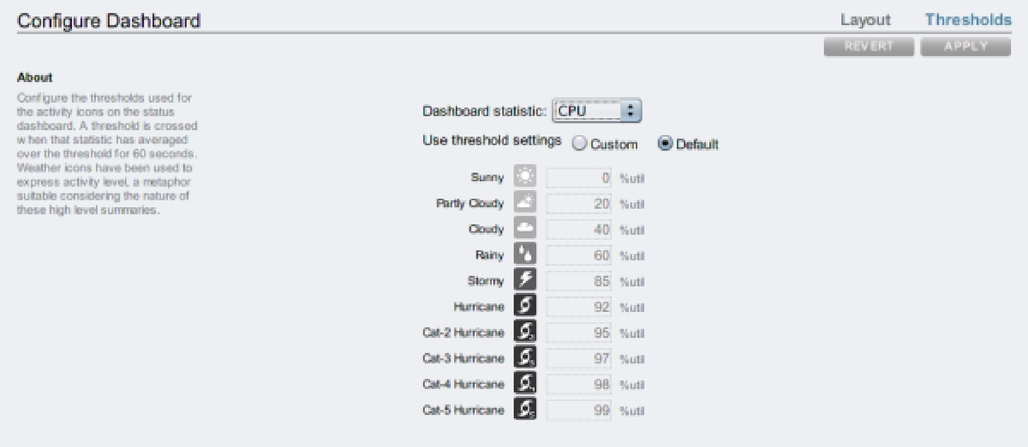
Name	Units	Description
Disk	operations/sec	Average number of operations to the physical storage devices.
HTTP	operations/sec	Average number of HTTP operations.
iSCSI	operations/sec	Average number of iSCSI operations.
FC	operations/sec	Average number of Fibre Channel operations.
Network	bytes/sec	Average 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 network traffic required to refresh the Dashboard, configure some of the activity graphs as "<empty>".

Thresholds

Use the Thresholds screen to configure the “dashboard activity” on page 48 weather icons. The defaults provided are based on heavy workloads, and may not be suitable for your environment.

FIGURE 2-7 Dashboard Activity Settings



Configure Dashboard

About

Configure the thresholds used for the activity icons on the status dashboard. A threshold is crossed w hen that statistic has averaged over the threshold for 60 seconds. Weather icons have been used to express activity level, a metaphor suitable considering the nature of these high level summaries.

Dashboard statistic: CPU

Use threshold settings ○ Custom ● Default

Sunny	0	%util
Partly Cloudy	20	%util
Cloudy	40	%util
Rainy	60	%util
Stormy	85	%util
Hurricane	92	%util
Cat-2 Hurricane	95	%util
Cat-3 Hurricane	97	%util
Cat-4 Hurricane	98	%util
Cat-5 Hurricane	99	%util

The weather icon that appears on the “Dashboard” on page 48 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.**

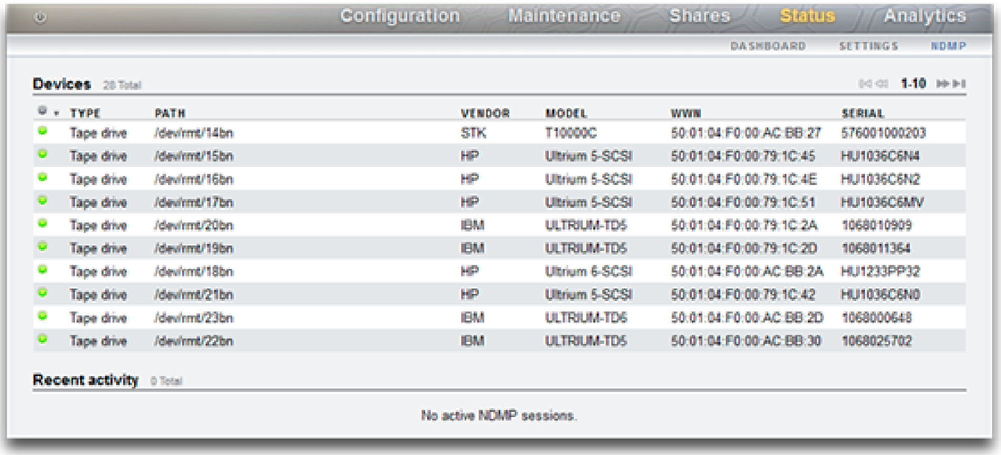
NDMP Status

When the “NDMP service” on page 221 has been configured and is active, the Status=>NDMP page shows the NDMP devices and recent client activity. A green indicator shows that the device is online and a gray indicator shows that the device is offline.

NDMP Status - BUI

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

FIGURE 2-8 NDMP Status BUI



Devices 28 Total

1-10

	TYPE	PATH	VENDOR	MODEL	WWN	SERIAL
●	Tape drive	/dev/rmt/14bn	STK	T10000C	50:01:04:F0:00:AC:BB:27	576001000203
●	Tape drive	/dev/rmt/15bn	HP	Ultrium 5-SCSI	50:01:04:F0:00:79:1C:45	HU1036C6N4
●	Tape drive	/dev/rmt/16bn	HP	Ultrium 5-SCSI	50:01:04:F0:00:79:1C:4E	HU1036C6N2
●	Tape drive	/dev/rmt/17bn	HP	Ultrium 5-SCSI	50:01:04:F0:00:79:1C:51	HU1036C6MV
●	Tape drive	/dev/rmt/20bn	IBM	ULTRIUM-TD5	50:01:04:F0:00:79:1C:2A	1068010909
●	Tape drive	/dev/rmt/19bn	IBM	ULTRIUM-TD5	50:01:04:F0:00:79:1C:2D	1068011364
●	Tape drive	/dev/rmt/18bn	HP	Ultrium 6-SCSI	50:01:04:F0:00:AC:BB:2A	HU1233PP32
●	Tape drive	/dev/rmt/21bn	HP	Ultrium 5-SCSI	50:01:04:F0:00:79:1C:42	HU1036C6N0
●	Tape drive	/dev/rmt/23bn	IBM	ULTRIUM-TD6	50:01:04:F0:00:AC:BB:2D	1068000648
●	Tape drive	/dev/rmt/22bn	IBM	ULTRIUM-TD5	50:01:04:F0:00:AC:BB:30	1068025702

Recent activity 0 Total

No active NDMP sessions.

NDMP Status - Devices

NDMP devices are listed here.

TABLE 2-5 NDMP Status - Devices

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

NDMP Status - Recent Activity

This section summarizes recent NDMP activity.

TABLE 2-6 NDMP Status - Recent Activity

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

NDMP Status - CLI

NDMP status is not currently available from the CLI.

♦ ♦ ♦ CHAPTER 3

Initial Configuration

Initial configuration consists of the following six sections.

- Chapter 4, "Network Configuration"
- "DNS" on page 254
- "Time" on page 258
- Name Services ("NIS" on page 236, "LDAP" on page 238, "Active Directory" on page 242)
- Chapter 5, "Storage Configuration"
- "Registration & Support" on page 261

Prerequisites

Initial system configuration is conducted after powering it on for the first time and establishing a connection, as documented in "Installation".

Note - Note that the option to perform initial configuration of a cluster is only available in the BUI. If electing this option, read Chapter 10, "Cluster Configuration" before beginning initial configuration for detailed additional steps that are required for successful cluster setup. Pay careful attention to the "Clustering Considerations for Networking" on page 166 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.

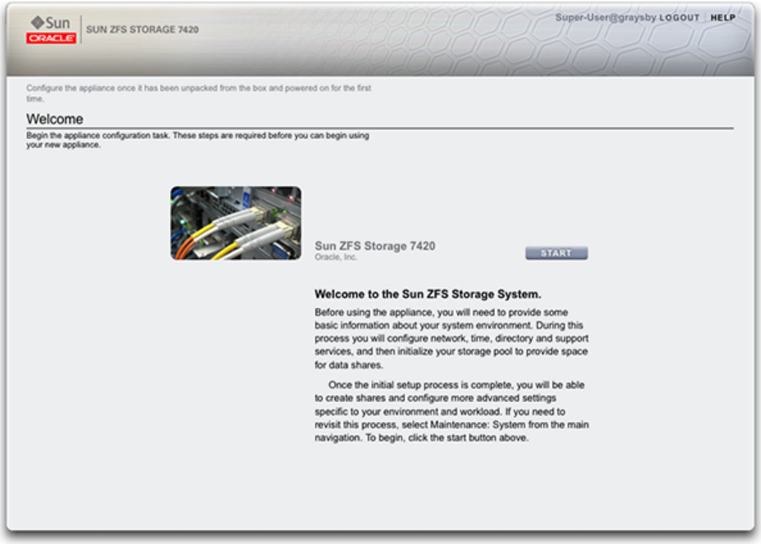
Performing Initial Configuration Using the BUI

Initial configuration configures network connectivity, several client network services, and the storage pool layout for standalone operation. When completed, the appliance is ready for use but does no shares configured for remote clients to access. To create shares or revisit settings, see Chapter 12, "Shares, Projects, and Schema".

Initial configuration can be repeated at a later time by clicking the "INITIAL SETUP" button on the “System” in “Oracle ZFS Storage Appliance Customer Service Manual ” screen or by entering the `maintenance system setup` context in the CLI.

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

FIGURE 3-1 ZFSSA Welcome Page



Configure the appliance once it has been unpacked from the box and powered on for the first time.

Welcome

Begin the appliance configuration task. These steps are required before you can begin using your new appliance.

Sun ZFS Storage 7420
Oracle, Inc.

Welcome to the Sun ZFS Storage System.

Before using the appliance, you will need to provide some basic information about your system environment. During this process you will configure network, time, directory and support services, and then initialize your storage pool to provide space for data shares.

Once the initial setup process is complete, you will be able to create shares and configure more advanced settings specific to your environment and workload. If you need to revisit this process, select Maintenance: System from the main navigation. To begin, click the start button above.

▼ Perform initial configuration

1. **To start initial configuration, on the Welcome page click Start.**
2. **For each page to commit your changes and go to the next screen, click Commit.**
3. **To go to a previous screen use the arrow buttons.**

Configuring Management Ports

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

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

Performing Initial Configuration Using the CLI

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

Login using the password you provided during "Installation" in "Oracle ZFS Storage Appliance Installation Guide ":

```
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 Chapter 4, "Network Configuration" section for additional documentation. Pay careful attention to the "Clustering Considerations for Networking" on page 166 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
       igb0 true     0:14:4f:8d:59:aa          1000 Mbit/s
       igb1 false    0:14:4f:8d:59:ab          0 Mbit/s
       igb2 false    0:14:4f:8d:59:ac          0 Mbit/s
       igb3 false    0:14:4f:8d:59:ad          0 Mbit/s

caji:maintenance system setup net> datalinks show
Datalinks:

   DATALINK CLASS          LINKS       LABEL
       igb0 device         igb0        Untitled Datalink

caji:maintenance system setup net> interfaces show
Interfaces:

  INTERFACE STATE  CLASS LINKS       ADDRS                 LABEL
       igb0 up     ip    igb0        192.168.2.80/22       Untitled Interface

caji:maintenance system setup net> done
```

Refer to the “DNS” on page 254 section for additional documentation about DNS.

```
Configure DNS. Configure the Domain Name Service.

Subcommands that are valid in this context:

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

   show                 => Show information pertinent to the current context

   commit               => Commit current state, including any changes

   abort                => Abort this task (potentially resulting in a
                           misconfigured system)
   done                    => Finish operating on "dns"

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

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

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

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

Configure Network Time Protocol (NTP) to synchronize the appliance time clock. See the "NTP" on page 258 section 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” on page 236, “LDAP” on page 238 and “Active Directory” on page 242 sections 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 Chapter 5, “Storage Configuration” section 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” on page 261) 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!
```

♦ ♦ ♦ CHAPTER 4

Network Configuration

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

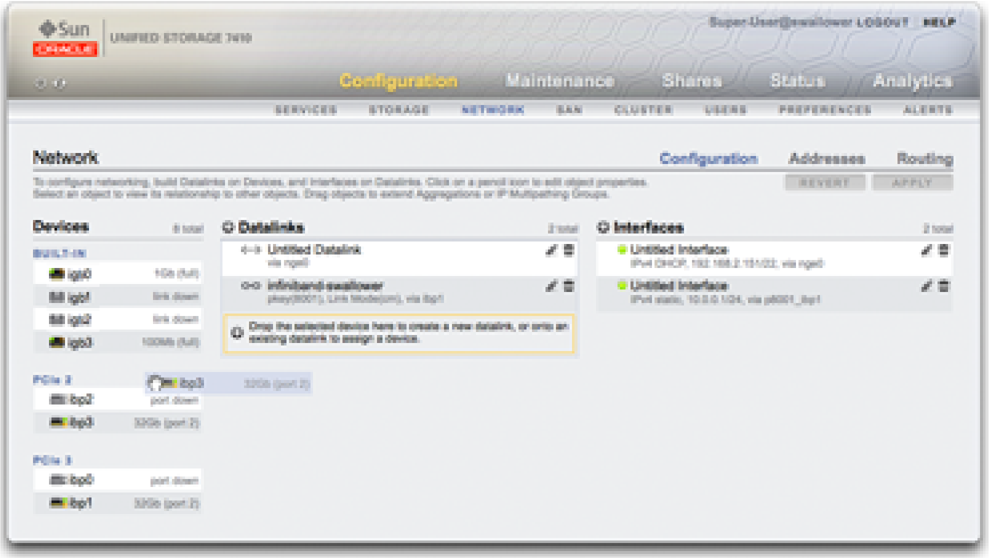
There are four components to a system's network configuration:

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

Network Configuration Page

In ZFSSA 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.

FIGURE 4-1 Network Configuration Window



To configure networking, build Datalinks on Devices, and Interfaces on Datalinks. Click on a pencil icon to edit object properties. Select an object to view its relationship to other objects. Drag objects to extend Aggregations or IP Multipathing Groups.

Drop the selected device here to create a new datalink, or onto an existing datalink to assign a device.

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

TABLE 4-1 Example - Single IP Address on a Single Port

Devices	Datalink	Interface
igb0	datalink1	deimos (192.168.2.80/22)

The following configuration is for a 3-way link aggregation:

TABLE 4-2 Example - Configuration for a 3-way Link Aggregation

Devices	Datalink	Interface
igb1, igb2, igb3	aggr1 (LACP aggregation)	phobos (192.168.2.81/22)

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

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:

- 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.
- VLANs - Virtual LANs to improve local network security and isolation. VLANs are recommended for administering the appliance; otherwise, use VNICs.
- VNICs - Virtual Network Interface Cards, which allow single or aggregated Ethernet datalinks to be split into multiple virtual (Ethernet) datalinks. VNICs can be optionally tagged with VLAN IDs, and can allow physical network port sharing in a cluster. Step-by-step instructions can be found in the "Clustering Considerations for Networking" on page 166 section below.

Note - VNIC-based and VLAN-based datalinks cannot share the same VLAN ID.

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

The following datalink settings are available:

TABLE 4-3 Datalink Settings

Property	Description
Name	Use the defined custom name. For example: "internal", "external", "adminnet", etc.
Speed	Use the defined speed. Valid values are auto, 10, 100, 1000 and 10000, representing autonegotiation, forced 10Mbit/sec, forced 100Mbit/sec, forced 1Gbit/sec and forced 10Gbit/sec. Speed and duplex must be either both forced to specific values or both set to autonegotiate. Not all networking devices support forcing to all possible

Property	Description
	speed/duplex combinations. Disabling autonegotiation is strongly discouraged. However, if the switch has autonegotiation disabled, it may be necessary to force speed (and duplex) to ensure the the datalink runs at the expected speed and duplex.
Duplex	Use the defined transmission direction. Valid CLI values are auto, half, and full, representing autonegotiation, half- and full-duplex respectively. Speed and duplex must be either both forced to specific values or both set to autonegotiate.
VLAN	Use VLAN headers.
VLAN ID	Use the defined VLAN identifier; optional for VNICs.
VNIC	Use a VNIC.
MTU	Use the defined maximum transmission unit (MTU) size. The default MTU is 1500 bytes. Specify a lower MTU (minimum 1280) to leave packet headroom (for example, for tunneling protocols). Specify a larger MTU (maximum 9000) to improve network performance. All systems and switches on the same LAN must be configured with the chosen MTU. After the MTU value is set and the new network configuration is committed to the system, you can return to the network screen and view the datalink status to see the exact MTU value in bytes that was selected. Note that a VLAN or VNIC cannot be configured with an MTU value larger than that of the underlying datalink.
LACP Aggregation	Use multiple network device LACP aggregation.
LACP Policy	Use the defined LACP policy for selecting an 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	Use the defined LACP communication mode. Active mode will send and receive LACP messages to negotiate connections and monitor the link status. Passive mode will listen for LACP messages only. Off mode will use the aggregated link but not detect link failure or switch configuration changes. Some network switch configurations, including Cisco Etherchannel, do not use the LACP protocol: the LACP mode should be set to "off" when using non-LACP aggregation in your network.
LACP Timer	Use the defined interval between LACP messages for Active mode.
IB Partition	Use IB Partitions.
Partition Key	Use the partition (fabric domain) in which the underlying port device is a member. The partition key (pkey) is

Property	Description
	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 "Network IP MultiPathing (IPMP) " on page 76 and Chapter 10, "Cluster Configuration" rules on the subnet manager.
IB Link Mode	Use the defined IB Link Mode. There are two modes: Unreliable Datagram and Connected. Unreliable Datagram lets a local queue pair communicate with multiple other queue pairs on any host and messages are communicated unacknowledged at the IB layer. Unreliable Datagram mode uses an MTU of 2044. Connected mode uses IB queue pairs and dedicates a local queue pair to communication with a dedicated remote queue pair. Connected mode uses an MTU of 65520 and can provides higher throughput than Unreliable Datagram.

Network Interfaces

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

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

The following interface settings are available:

TABLE 4-4 Interface Settings

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.

Property	Description
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)
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

Network IP MultiPathing (IPMP)

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

Each IP interface in an IPMP group is designated either <i>active</i> or <i>standby</i>:

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

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

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

If probe-based failure detection is enabled on an IP interface, the system will determine which target systems to probe dynamically. First, the routing table will be scanned for gateways (routers) on the same subnet as the IP interface's test address and up to five will be selected. If no gateways on the same subnet were found, the system will send a multicast ICMP probe (to 224.0.01. 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.

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

The system will probe selected target systems in round-robin fashion. If five consecutive probes are unanswered, the IP interface will be considered failed. Conversely, if ten consecutive probes are answered, the system will consider a previously failed IP interface as repaired. You can set the system's IPMP probe failure detection time from the “IPMP” on page 257 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 are here “Network IP MultiPathing (IPMP) ” on page 76.

For information about private local interfaces, see Chapter 10, “Cluster Configuration”.

Network Performance and Availability

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

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

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

Network Routing Configuration

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.

Network Routing Entries

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

TABLE 4-5 Routing Entry 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

Field	Description	Examples
Interface	IP interface the packet will be sent on	igb0

A routing entry with a "destination" field of `0.0.0.0/0` matches any packet (if no other route matches more precisely), and is thus known as a 'default' route. In the BUI, default routes are distinguished from non-default routes by an additional property:

TABLE 4-6 Distinguishing Default from Non-default Routes

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:

TABLE 4-7 Supported Route Types

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:

TABLE 4-8 Unavailable Static Route Type

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

Network Routing Properties

TABLE 4-9 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:

TABLE 4-10 Multihoming Policies

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.
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

Policy	Description
	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.

Network Configuration Using the BUI

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

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

TABLE 4-11 Network Configuration Icons

icon	description
⊕	Add new datalink/interface/route
✎	Edit datalink/interface/route settings
⊘	Editing disabled
🗑	Destroy datalink/interface/route

icon	description
🚫	Destruction disabled
✥	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 or VNIC
{⁞⁞⁞}	network datalink aggregation
{∘∘∘}	network datalink aggregation VLAN or VNIC
🔗	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.

Network Configuration Page

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.

Network Addresses

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

TABLE 4-12 Summary of the Current Network Configuration

Field	Description	Example
Network Datalink	Datalink name and detail summary	datalink1 (via igb0)
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

Network Routing Page

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

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

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

Network Configuration Using the CLI

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

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

DEVICE      UP     SPEED         MAC
igb0        true   1000 Mbit/s   0:14:4f:9a:b9:0
igb1        true   1000 Mbit/s   0:14:4f:9a:b9:1
igb2        true   1000 Mbit/s   0:14:4f:9a:b8:fe
igb3        true   1000 Mbit/s   0:14:4f:9a:b8:ff

caji:configuration net> datalinks show
Datalinks:

   DATALINK CLASS          LINKS       LABEL
       igb0 device         igb0        datalink1

caji:configuration net> interfaces show
Interfaces:

  INTERFACE STATE  CLASS LINKS       ADDRS                  LABEL
       igb0 up     ip    igb0        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     igb0      dhcp
route-001  192.168.0.0/22                  192.168.2.142   igb0      system
```

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

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

  show                 => Show information pertinent to the current context

  commit               => Commit current state, including any changes

  abort                => Abort creation of "vnic"

  done                 => Finish operating on "vnic"

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

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

  available            => Get values that can be assigned to the links
                          parameter when creating a network component.
```

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

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

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

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

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

caji:configuration net interfaces> ip
caji:configuration net interfaces ip (uncommitted)> available
aggr2,aggr1

caji:configuration net interfaces> ipmp
caji:configuration net interfaces ipmp (uncommitted)> available
vnic4,vnic3,igb5,igb4
```

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

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

   DATALINK CLASS          LINKS       LABEL
       igb0 device         igb0        datalink1
       igb1 device         igb1        datalink2

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

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

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

```
caji:configuration net routing> create
caji:configuration net route (uncommitted)> set family=IPv4
                   family = IPv4 (uncommitted)
caji:configuration net route (uncommitted)> set destination=0.0.0.0
                   destination = 0.0.0.0 (uncommitted)
caji:configuration net route (uncommitted)> set mask=0
                   mask = 0 (uncommitted)
caji:configuration net route (uncommitted)> set interface=igb1
                   interface = igb1 (uncommitted)
caji:configuration net route (uncommitted)> set gateway=10.0.1.2
                   gateway = 10.0.1.2 (uncommitted)
caji:configuration net route (uncommitted)> commit
```

Network Configuration Tasks Using the BUI

▼ Creating a single port interface

1. **Click the Datalinks ⊕ icon.**
2. **Optionally set name and select custom MTU radio button (typing 9000 in the text box).**
3. **Choose a device from the Devices list.**
4. **Click "APPLY". The datalink will appear in the Datalinks list.**
5. **Click the Interface ⊕ icon.**
6. **Set desired properties, and choose the datalink previously created.**
7. **Click "APPLY". The interface will appear in the Interfaces list.**
8. **The running appliance network configuration has not yet changed. When you are finished configuring interfaces, click "APPLY" at the top to commit the configuration.**

▼ Modifying an interface

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 MTU.**
4. **Click "APPLY".**
5. **Now Drag the datalink over to the Interfaces list.**
6. **Set desired properties, and click "APPLY".**
7. **Click "APPLY" at the top of the screen to commit the configuration.**

▼ Creating an LACP aggregated link interface

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.**

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

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

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

2. **Click the Interface ⊕ icon.**
3. **Optionally change the name of the interface.**
4. **Click the IP MultiPathing Group check box.**
5. **Click the Use IPv4 Protocol or/and the Use IPv6 Protocol and specify the IP addresses for the IPMP interface.**
6. **Choose the interfaces created in the fist step from the Interfaces list.**
7. **Set each chosen interface to be either "Active" or "Standby", as desired.**
8. **Click "APPLY".**

▼ Creating an IPMP group using link-state only failure detection

1. **Create one or more "underlying" IP interfaces with the IP address 0.0.0.0/8 to be used as the components of the IPMP group (see separate task to create a single-port interfaces above).**
2. **Click the Interface ⊕ icon.**
3. **Optionally change the name of the interface.**
4. **Click the IP MultiPathing Group check box.**
5. **Click the Use IPv4 Protocol or/and the Use IPv6 Protocol and specify the IP addresses for the IPMP interface.**
6. **Choose the interfaces created in the first step from the Interfaces list.**
7. **Set each chosen interface to be either "Active" or "Standby", as desired.**
8. **Click "APPLY".**

▼ Extending an LACP aggregation

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

2. **Click the ✥ icon, and drag the device onto an aggregation datalink, and release.**
3. **Click "APPLY" at the top of the page to commit this configuration.**

Extending an IPMP group

1. **Mouse-over an interface in the Interfaces list.**
2. **Click the ✥ icon, and drag the device onto an IPMP interface, and release.**
3. **Click "APPLY" at the top of the page to commit this configuration.**

Creating an InfiniBand partition datalink and interface

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.**

Creating a VNIC without a VLAN ID for clustered controllers

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

1. **When the cluster is in state AKCS_CLUSTERED, click the Datalinks ⊕ icon.**
2. **Optionally set name and MTU.**
3. **Choose a device from the Devices list and click "APPLY". The datalink appears in the Datalinks list.**
4. **Click the Interface ⊕ icon.**
5. **Set desired properties, choose the datalink previously created, and click "APPLY". The interface appears in the Interfaces list.**
6. **Click the Datalinks ⊕ icon.**
7. **Select the VNIC checkbox, optionally set name and MTU (equal to or less than the value in step 2), and click "APPLY". The new VNIC datalink appears in the Datalinks list.**
8. **Click the Interface ⊕ icon.**
9. **Set desired properties, choose the VNIC datalink previously created, and click "APPLY". The interface appears in the Interfaces list.**
10. **The running appliance network configuration has not yet changed. When you are finished configuring interfaces, click "APPLY" at the top to commit the configuration.**
11. **Click the Cluster tab. The two newly created interfaces appear in the Resource section with default owners.**
12. **Use the Owner pull-down list to assign one of the two interfaces to the other head and click "APPLY".**

Creating VNICs with the same VLAN ID for clustered controllers

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

1. **When the cluster is in state AKCS_CLUSTERED, click the Datalinks ⊕ icon.**
2. **Select the VNIC checkbox, optionally set name and MTU, set the VLAN ID, choose a device from the Devices list, and click "APPLY". The new VNIC datalink appears in the Datalinks list.**
3. **Click the Interface ⊕ icon.**
4. **Set desired properties, choose the VNIC datalink previously created, and click "APPLY". The interface appears in the Interfaces list.**
5. **Create another VNIC as described in steps 1 and 2 with the same Device and VLAN ID, and create an interface for it as described in steps 3 and 4.**
6. **The running appliance network configuration has not yet changed. When you are finished configuring interfaces, click "APPLY" at the top to commit the configuration.**
7. **Click the Cluster tab. The two newly created interfaces appear in the Resource section with default owners.**
8. **Use the Owner pull-down list to assign one of the two interfaces to the other head and click "APPLY".**

Adding a static route

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.**

Network Configuration Tasks Using the CLI

▼ Adding a static route

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

▼ Deleting a static route

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

▼ Changing the multihoming property to strict

1. **Go to `configuration net routing`**
2. **Enter `set multihoming=strict`**

3. **Enter `commit`**

♦ ♦ ♦ CHAPTER 5

Storage Configuration

Storage is configured in pools that are characterized by their underlying data redundancy, and provide space that is shared across all filesystems and LUNs. More information about how storage pools relate to individual filesystems or LUNs can be found in the "Shares section" on page 280.

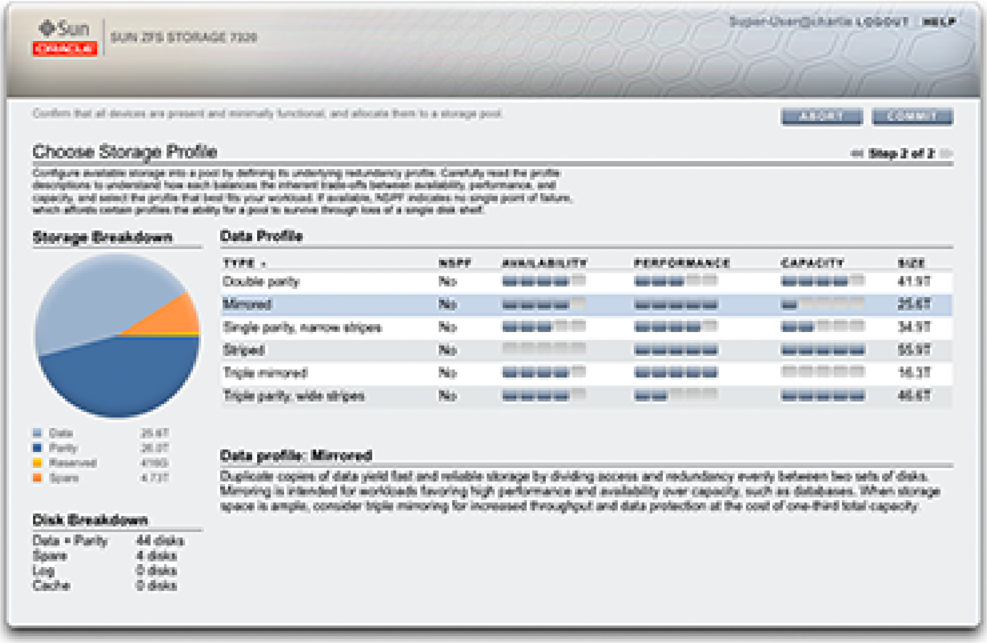
Each node can have any number of pools, and each pool can be assigned ownership independently in a cluster. While arbitrary number of pools are supported, creating multiple pools with the same redundancy characteristics owned by the same cluster head is not advised. Doing so will result in poor performance, 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 over provisioning by approximately 20%. The "Shares UI" on page 280 can be used to determine how much space is currently being used.

Storage Configuration Profile

FIGURE 5-1 Storage Configuration Profile



Confirm that all devices are present and minimally functional, and allocate them to a storage pool.

Choose Storage Profile

Configure available storage into a pool by defining its underlying redundancy profile. Carefully read the profile descriptions to understand how each balances the inherent trade-offs between availability, performance, and capacity, and select the profile that best fits your workload. If available, NSPF indicates no single point of failure, which affords certain profiles the ability for a pool to survive through loss of a single disk shelf.

Storage Breakdown

▨ Data	25.6T
■ Parity	26.0T
■ Reserved	416G
■ Spare	4.73T

Disk Breakdown

Data + Parity	44 disks
Spare	4 disks
Log	0 disks
Cache	0 disks

Data Profile

TYPE	NSPF	AVAILABILITY	PERFORMANCE	CAPACITY	SIZE
Double parity	No	■■■■□	■■■□□	■■■■□	41.5T
Mirrored	No	■■■■□	■■■■■	■□□□□	25.6T
Single parity, narrow stripes	No	■■■□□	■■■■□	■■□□□	34.5T
Striped	No	□□□□□	■■■■■	■■■■■	55.9T
Triple mirrored	No	■■■■□	■■■■■	□□□□□	16.3T
Triple parity, wide stripes	No	■■■■□	■■□□□	■■■■■	46.6T

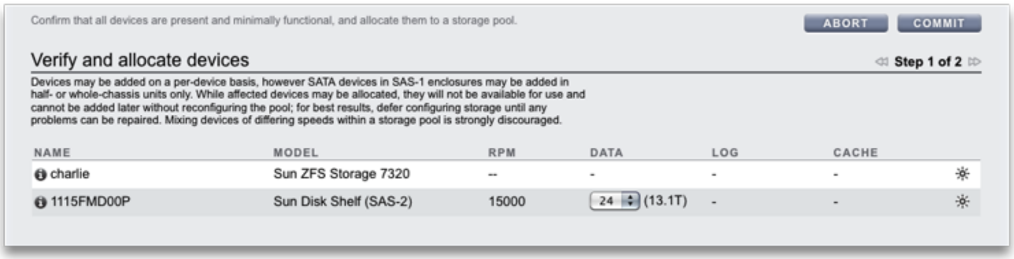
Data profile: Mirrored

Duplicate copies of data yield fast and reliable storage by dividing access and redundancy evenly between two sets of disks. Mirroring is intended for workloads favoring high performance and availability over capacity, such as databases. When storage space is ample, consider triple mirroring for increased throughput and data protection at the cost of one-third total capacity.

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.

FIGURE 5-2 Verify and Allocate Devices



Confirm that all devices are present and minimally functional, and allocate them to a storage pool.

Verify and allocate devices

Devices may be added on a per-device basis, however SATA devices in SAS-1 enclosures may be added in half- or whole-chassis units only. While affected devices may be allocated, they will not be available for use and cannot be added later without reconfiguring the pool; for best results, defer configuring storage until any problems can be repaired. Mixing devices of differing speeds within a storage pool is strongly discouraged.

NAME	MODEL	RPM	DATA	LOG	CACHE	
charlie	Sun ZFS Storage 7320	--	-	-	-	☼
1115FMD00P	Sun Disk Shelf (SAS-2)	15000	24 (13.1T)	-	-	☼

Storage Configuration Rules and Guidelines

For optimal performance, keep in mind the following:

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

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

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

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

Storage Verification

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

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

You can select the following:

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

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

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

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

Storage Allocation on SAS-2 Systems

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

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

Data Profile Configuration

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

TABLE 5-1 Data Profile Configuration

Data Profile	Description
Dual Parity Options	
Triple mirrored	Data is triply mirrored, yielding a very highly reliable and high-performing system (for example, storage for a critical database). This configuration is intended

Data Profile	Description
	for situations in which maximum performance and availability are required. Compared with a two-way mirror, a three-way mirror adds additional IOPS per stored block and higher level protection against failures. Note: A controller without expansion storage should not be configured with triple mirroring.
Double parity RAID	RAID in which each stripe contains two parity disks. As with triple mirroring, this yields high availability, as data remains available with the failure of any two disks. Double parity RAID is a higher capacity option than the mirroring options and is intended either for high-throughput sequential-access workloads (such as backup) or for storing large amounts of data with low random-read component.
Single Parity Options	
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).
Single parity RAID, narrow stripes	RAID in which each stripe is kept to three data disks and a single parity disk. For situations in which single parity protection is acceptable, single parity RAID offers a much higher capacity option than simple mirroring. This higher capacity needs to be balanced against a lower random read capability than mirrored options. Single parity RAID can be considered for non-critical applications with a moderate random read component. For pure streaming workloads, give preference to the Double parity RAID option which has higher capacity and more throughput.
Other	
Striped	Data is striped across disks, with no redundancy. While this maximizes both performance and capacity, a single disk failure will result in data loss. This configuration is not recommended. For pure streaming workloads, consider using Double parity RAID.
Triple parity RAID, wide stripes	RAID in which each stripe has three disks for parity. This is the highest capacity option apart from Striped Data. Resilvering data after one or more drive failures can take significantly longer due to the wide stripes and low random I/O performance. As with other RAID configurations, the presence of cache can mitigate the effects on read performance. This configuration is not generally recommended.

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

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

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

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

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

Importing Existing Storage Pools

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.

Adding Additional Storage

Use this action to add additional storage to your existing pool. The verification step is identical to the verification step during initial configuration. The storage must be added using the same profile that was used to configure the pool initially. If there is insufficient storage to configure the system with the current profile, some attributes can be sacrificed. For example, adding a single 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.

Unconfiguring Storage

This removes any active filesystems and LUNs and unconfigure the storage pool, making the raw storage available for future storage configuration. This process can be undone by importing the unconfigured storage pool, provided the raw storage has not since been used as part of an active storage pool.

Storage Pool Scrub

This initiates 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.

Configuring Storage Using the BUI

▼ Configuring a Storage Pool

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

1. **Click the ⊕ button above the list of storage pools**

2. **Enter a name for the storage pool**

3. **At the "Allocate and verify storage" screen, configure the 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".**

Adding Cache Devices to an Existing Pool

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

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

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

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

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

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

Configuring Storage Using the CLI

Adding Cache Devices to an Existing Pool

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

2. **At the command line, enter:**
3. : `poc:> configuration storage`
4. **Specify the pool you want to add the device to:**
5. : `poc:configuration storage (pool_2)> set pool=pool_2`
6. : `pool = pool_2`
7. : `poc:configuration storage (pool_2)> add`
8. **:A message reminds you to verify that the device is correctly installed. Note that mixing device types and speeds is strongly discouraged.**
9. **Show the device information for the pool:**
10. : `poc:configuration storage (pool_2) verify> show`
11. : `ID STATUS ALLOCATION DATA LOG CACHE RPM`
12. : `0 ok custom 0 0 0/4 1.86T`
13. : `1 ok custom 0 0/2 34G 0 15000`
14. : `2 ok custom 0 0/2 34G 0 15000`
15. **Specify which disk shelf and the number of Logzillas or Readzillas to use. In the following example, `1-log=1` allocates one Logzilla from the first disk shelf.**
16. : `poc:configuration storage (pool_2) verify> set 1-log=1`
17. : `1-log = 1`
18. **:Note: A value of "1-log=2" would allocate two Logzillas from the first disk shelf.**
19. **:This example allocates one Readzilla from the first disk shelf.**
20. : `poc:configuration storage (pool_2) verify> set 1-cache=1`
21. : `1-cache = 1`
22. **Enter `done`.**
23. : `poc:configuration storage (pool_2) verify> done`

24. **: Note: If you add an odd number of Logzilla devices to a pool, or if a pool does not have already a profile, enter `set log_profile=log_mirror` to set the log profile.**

25. **Enter `show` to display the profile.**

26. : `poc:configuration storage (pool_2) config> show`

27. :

28. : `PROFILE CAPCTY NSPF DESCRIPTION`

29. : `log_profile = log_stripe 17G no Striped log`

30. **Enter `done` to complete the task:**

31. : `poc:configuration storage (pool_2) config> done`

32. : `poc:configuration storage (pool_2)>`

♦ ♦ ♦ CHAPTER 6

Storage Area Network Configuration

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

- A client 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.

SAN Targets and Initiators

Targets and initiators are configured by protocol. Refer to the documentation on a particular protocol ("SAN Fibre Channel" on page 109, "iSCSI" on page 200 or "SRP" on page 127) for details.

SAN Target and Initiator Groups

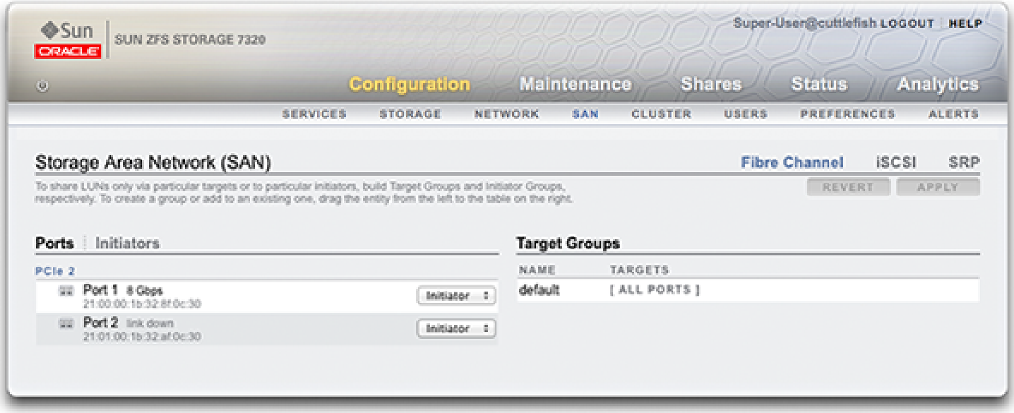
Target and initiator groups define sets of targets and initiators that can be associated with LUNs. A LUN that is associated with a target group can only be seen via the targets in the group. If a LUN is not explicitly associated with a target group, it is in the *default target group* and will be accessible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

To avoid possible LUN conflicts when an initiator belongs to multiple groups, configure initiators within all groups before associating groups with LUNs.

Configuring SAN Using the BUI

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

FIGURE 6-1 SAN BUI Page



Storage Area Network (SAN)

To share LUNs only via particular targets or to particular initiators, build Target Groups and Initiator Groups, respectively. To create a group or add to an existing one, drag the entity from the left to the table on the right.

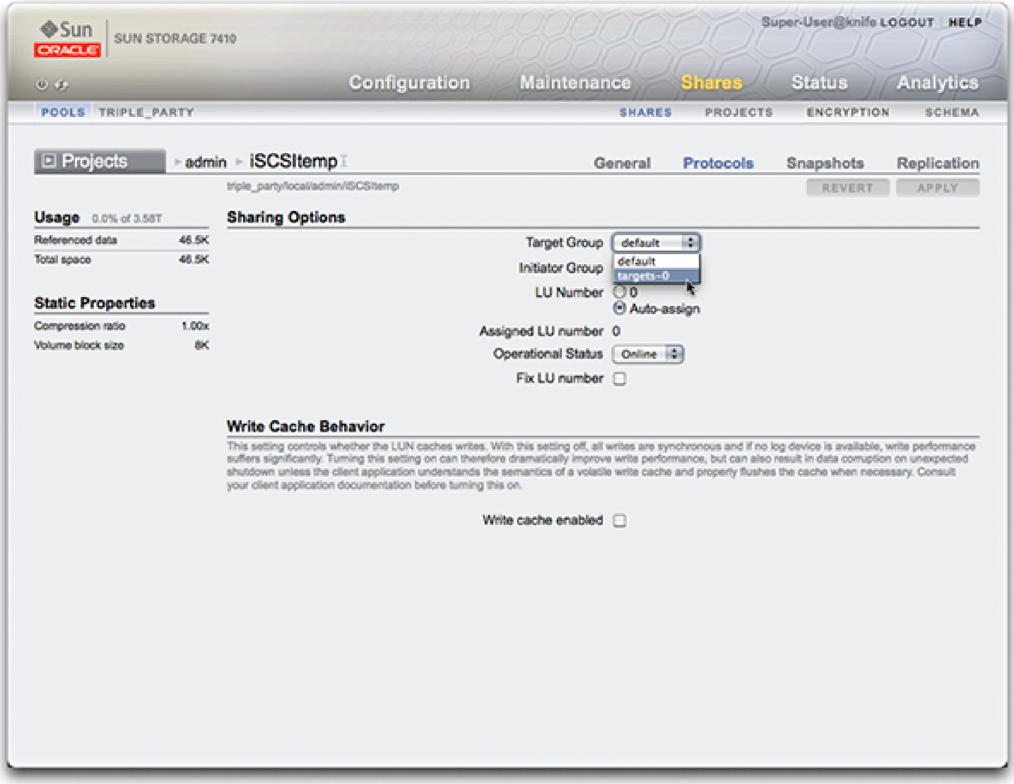
PCIe 2

Port 1 8 Gbps	21:00:00:1b:32:8f:0c:30	Initiator
Port 2 link down	21:01:00:1b:32:af:0c:30	Initiator

NAME	TARGETS
default	[ALL PORTS]

To associate a LUN, go to the Shares > Shares > Protocols page and then configure the Target Group and Initiator Group controls.

FIGURE 6-2 Associate a LUN



triple_party/local/admin/iSCSItemp

Usage 0.0% of 3.58T

Referenced data	46.5K
Total space	46.5K

Static Properties

Compression ratio	1.00x
Volume block size	8K

Sharing Options

Target Group: default (default, targets-0)

Initiator Group

LU Number: ○ 0 ● Auto-assign

Assigned LU number: 0

Operational Status: Online

Fix LU number: □

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.

Write cache enabled: □

Configuring SAN Using the 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.

SAN Terminology

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

TABLE 6-1 SAN Terminology

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.

Term	Description
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.
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.
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 administrator, and is identified by unique addressing methods. A target, once configured, consists of zero or more logical units.
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.

Each LUN has several properties which control how the volume is exported. See the “Protocols” on page 311 section for more information.

SAN Fibre Channel

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

FC Port Target Configuration

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

Each FC port is assigned a World Wide Name (WWN), and -- as with other block protocols -- FC targets may be grouped into "SAN Target and Initiator Groups" on page 105, allowing port bandwidth to be dedicated to specific LUNs or groups of LUNs. Once an FC port is configured as a target, the remotely discovered ports can be examined and verified.

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

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.

FC 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.

Clustering Considerations

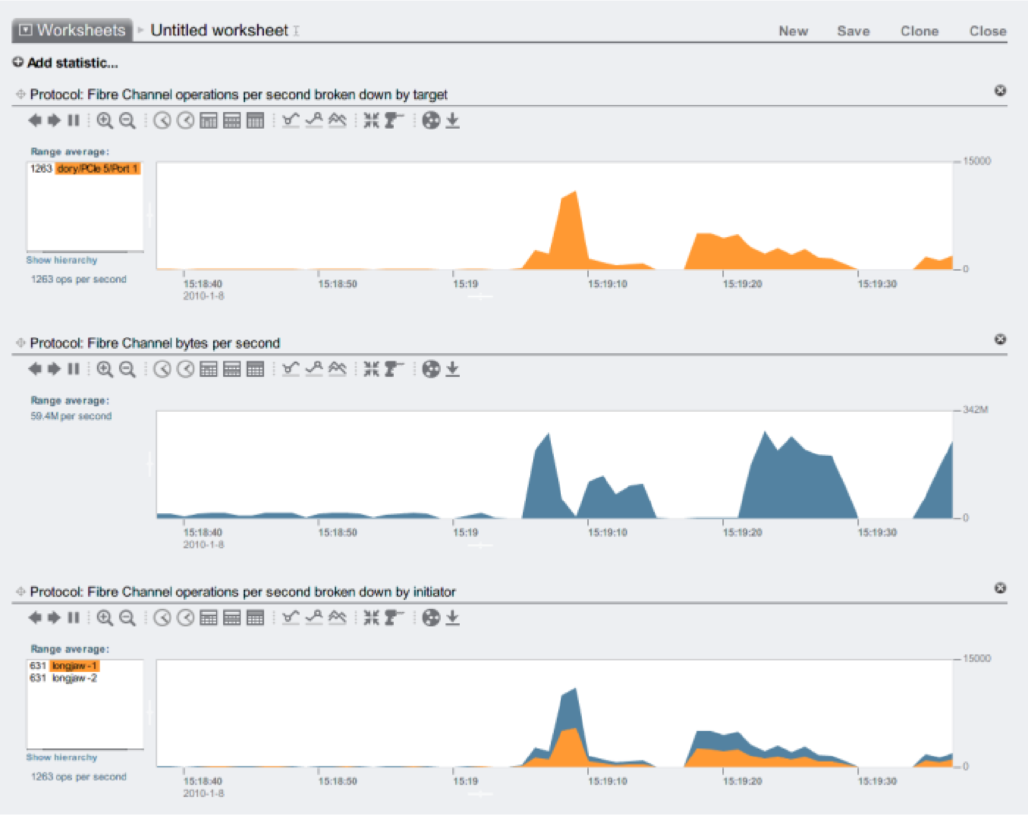
The appliance is an ALUA-compliant array. Properly configuring an FC initiator in an ALUA environment requires an ALUA-aware driver, and may require initiator-specific tuning. See

"Oracle ZFS Storage Appliance: How to set up Client Multipathing" (Doc ID 1628999.1) for more information.

Performance Considerations

FC performance can be observed via “Analytics” in “Oracle ZFS Storage Appliance Analytics Guide ”, whereby one can breakdown operations or throughput by initiator, target, or LUN:

FIGURE 6-3 FC Performance



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

Troubleshooting FC

FC 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 7420 dual headed cluster with one head having 2 FC ports and 100 LUNs and the other head having 4 FC ports and 28 LUNs, one should take the pessimal maximum queue depth to be two ports times 2048 commands divided by 100 LUNs plus 28 LUNs -- or 32 commands per LUN.

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

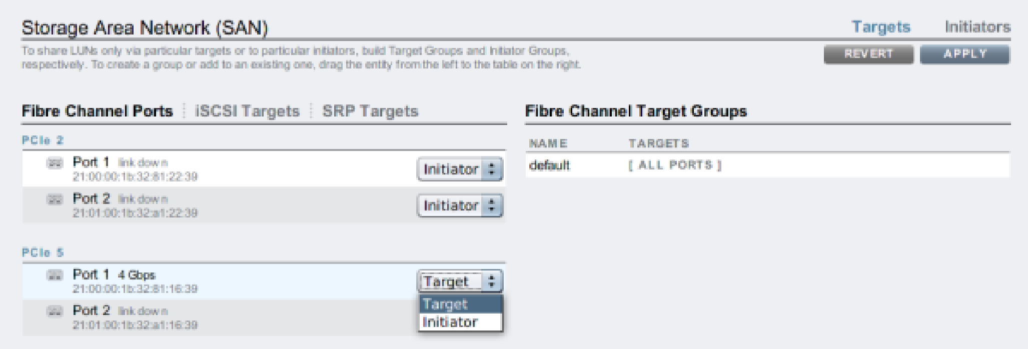
FC Link-level Issues

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

Configuring FC Using the 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.



To share LUNs only via particular targets or to particular initiators, build Target Groups and Initiator Groups, respectively. To create a group or add to an existing one, drag the entity from the left to the table on the right.

PCIe 2

Port 1 link down	21:00:00:1b:32:81:22:39	Initiator
Port 2 link down	21:01:00:1b:32:a1:22:39	Initiator

PCIe 5

Port 1 4 Gbps	21:00:00:1b:32:81:16:39	Target
Port 2 link down	21:01:00:1b:32:a1:16:39	

NAME	TARGETS
default	[ALL PORTS]

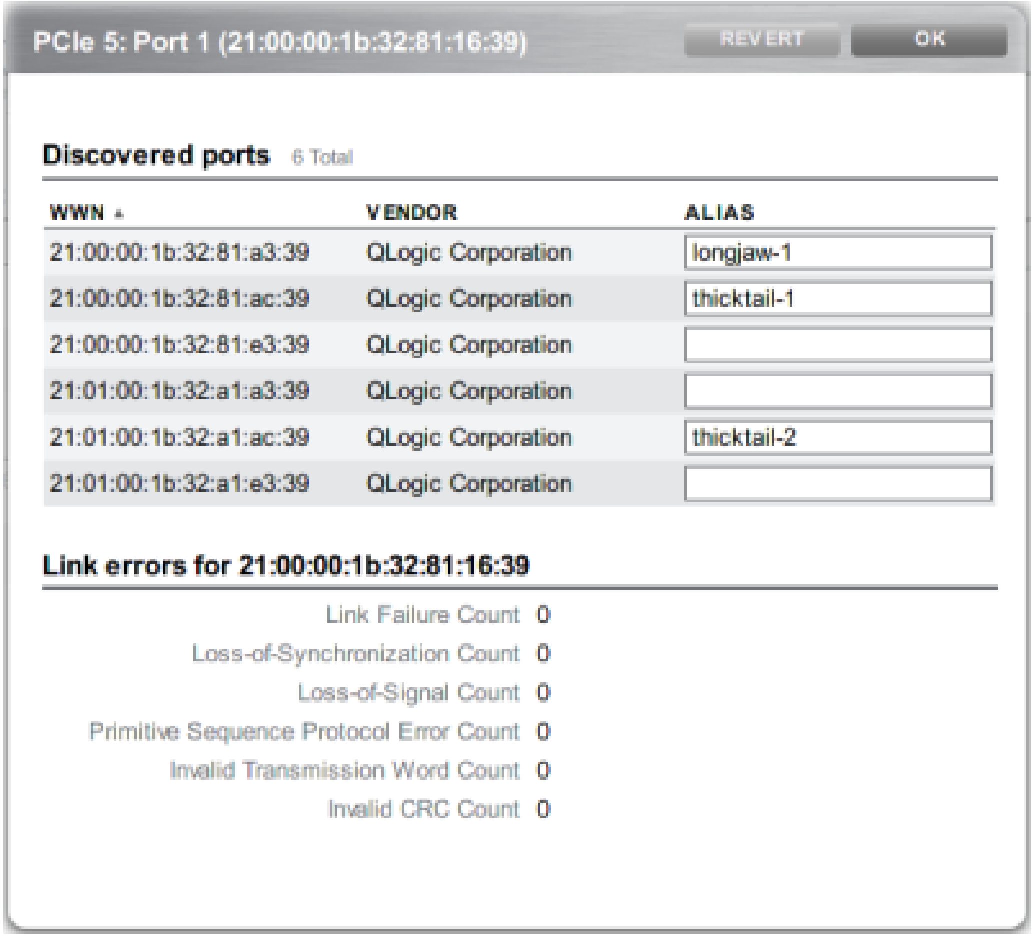
After setting desired ports to Target, click the Apply button. A confirmation message will appear notifying you that the appliance will reboot immediately. Confirm that you want to reboot.

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 ℹ 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.

FIGURE 6-4 Discovered FC Ports



Discovered ports 6 Total

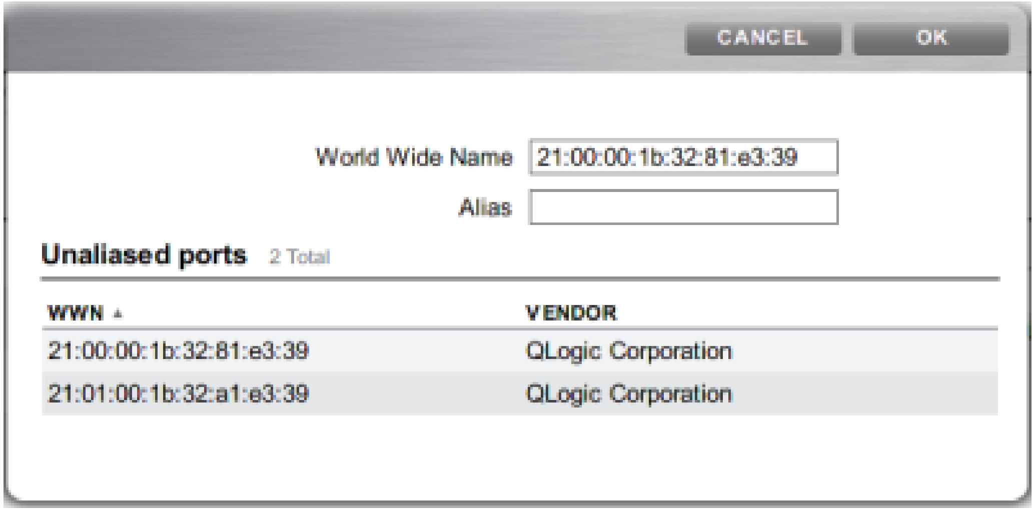
WWN ▲	VENDOR	ALIAS
21:00:00:1b:32:81:a3:39	QLogic Corporation	longjaw-1
21:00:00:1b:32:81:ac:39	QLogic Corporation	thicktail-1
21:00:00:1b:32:81:e3:39	QLogic Corporation	
21:01:00:1b:32:a1:a3:39	QLogic Corporation	
21:01:00:1b:32:a1:ac:39	QLogic Corporation	thicktail-2
21:01:00:1b:32:a1:e3:39	QLogic Corporation	

Link errors for 21:00:00:1b:32:81:16:39

Link Failure Count	0
Loss-of-Synchronization Count	0
Loss-of-Signal Count	0
Primitive Sequence Protocol Error Count	0
Invalid Transmission Word Count	0
Invalid CRC Count	0

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.



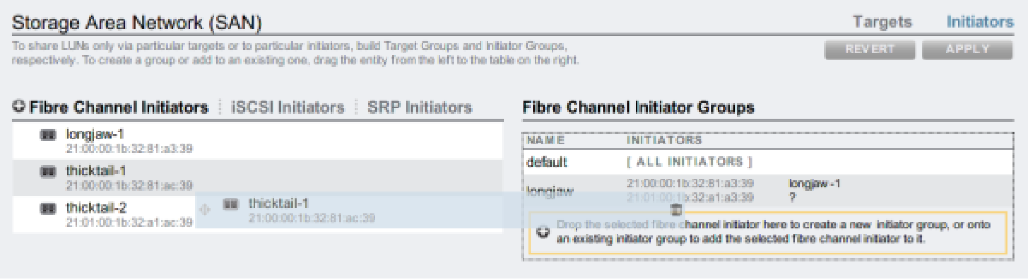
World Wide Name: 21:00:00:1b:32:81:e3:39
Alias:

Unaliased ports 2 Total

WWN ▲	VENDOR
21:00:00:1b:32:81:e3:39	QLogic Corporation
21:01:00:1b:32:a1:e3:39	QLogic Corporation

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

FIGURE 6-5 FC Initiator Groups List



Storage Area Network (SAN)

To share LUNs only via particular targets or to particular initiators, build Target Groups and Initiator Groups, respectively. To create a group or add to an existing one, drag the entity from the left to the table on the right.

Fibre Channel Initiators

- longjaw-1 21:00:00:1b:32:81:a3:39
- thicktail-1 21:00:00:1b:32:81:ac:39
- thicktail-2 21:01:00:1b:32:a1:ac:39
- thicktail-1 21:00:00:1b:32:81:ac:39

Fibre Channel Initiator Groups

NAME	INITIATORS	
default	[ALL INITIATORS]	
longjaw	21:00:00:1b:32:81:a3:39 21:01:00:1b:32:a1:a3:39	longjaw-1 ?

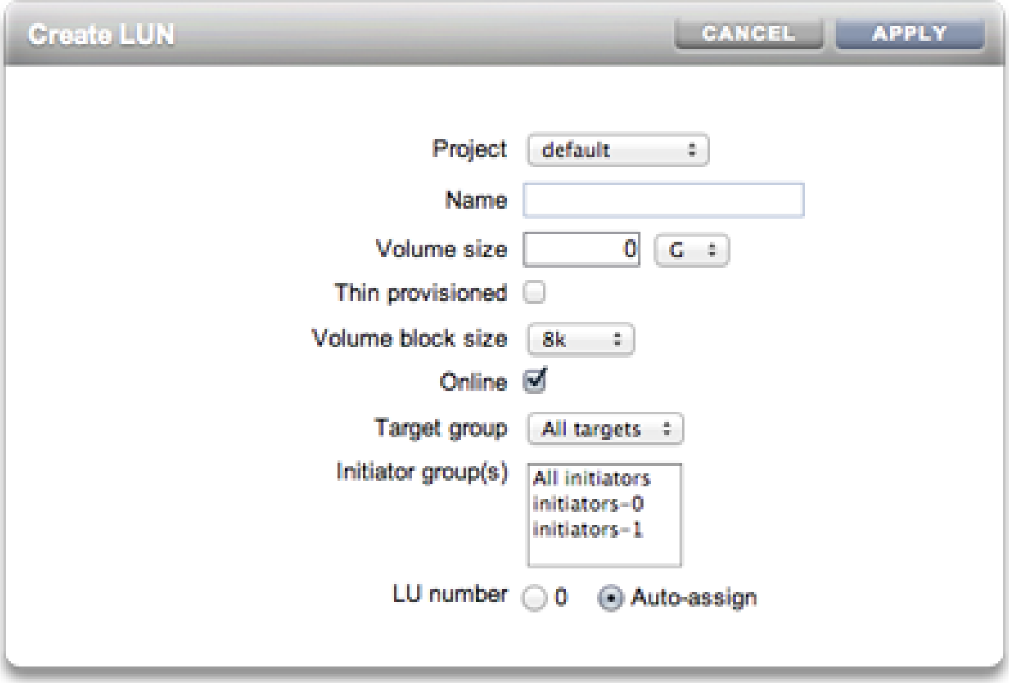
Drop the selected fibre channel initiator here to create a new initiator group, or onto an existing initiator group to add the selected fibre channel initiator to it.

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.

FIGURE 6-6 Associating a LUN with an FC Initiator Group



Project: default

Name:

Volume size: 0 G

Thin provisioned □

Volume block size: 8k

Online ☑

Target group: All targets

Initiator group(s):
All initiators
initiators-0
initiators-1

LU number ○ 0 ● Auto-assign

Configuring FC Using the CLI

Changing Modes of FC Ports

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

Viewing Discovered FC Ports

```
dory:configuration san fc targets> 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 fc targets> select target-000
dory:configuration san fc targets target-000> show
Properties:
                          wwn = wwn.2100001B32811639
                         port = PCIe 5: Port 1
                         mode = target
                        speed = 4 Gbit/s
             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 fc initiators> create
dory:configuration san fc initiators (uncommitted)> set name=lefteye
dory:configuration san fc initiators (uncommitted)>
    set initiators=wwn.2101001B32A1AC39,wwn.2100001B3281AC39
dory:configuration san fc initiators (uncommitted)> commit
dory:configuration san fc initiators> list
GROUP     NAME
group-001 lefteye
         |
         +-> INITIATORS
             wwn.2101001B32A1AC39
             wwn.2100001B3281AC39
```

Associating a LUN with an FC initiator group

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" on page 36 and "Simple CLI Scripting and Batching Commands" on page 36 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 ..');
}
```

iSCSI

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.

Note: Multiple connections per session is not supported.

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

TABLE 6-2 iSCSI Target Properties

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:

TABLE 6-3 Target Status Icons

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.

TABLE 6-4 iSCSI Initiator 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.

Troubleshooting iSCSI

For tips on troubleshooting common iSCSI misconfiguration, see the "iSCSI" on page 200 section.

Observing iSCSI Performance

iSCSI performance can be observed via "Analytics" in "Oracle ZFS Storage Appliance Analytics Guide ", whereby one can breakdown operations or throughput by initiator, target, or LUN.

Configuring iSCSI Using the BUI

▼ Creating an Analytics Worksheet

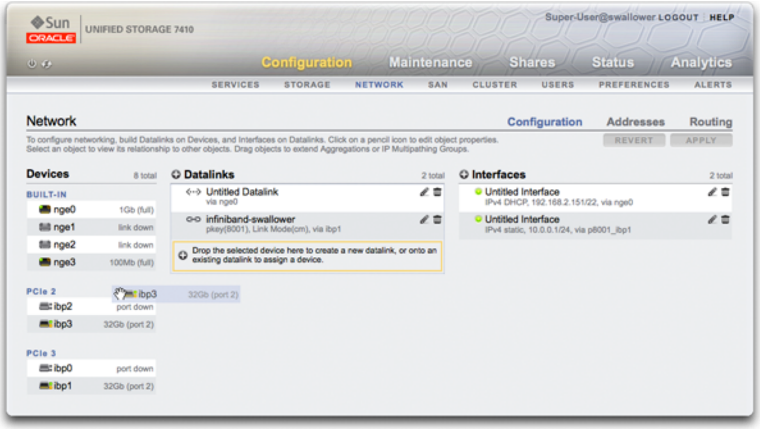
To create an analytics worksheet for observing operations by initiator, complete the following:

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.**

▼ iSER Target Configuration

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

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



Network

To configure networking, build Datalinks on Devices, and Interfaces on Datalinks. Click on a pencil icon to edit object properties. Select an object to view its relationship to other objects. Drag objects to extend Aggregations or IP Multipathing Groups.

Devices 8 total

BUILT-IN

nge0	1Gb (full)
nge1	link down
nge2	link down
nge3	100Mb (full)

PCIe 2

ibp3	32Gb (port 2)
ibp2	port down
ibp3	32Gb (port 2)

PCIe 3

ibp0	port down
ibp1	32Gb (port 2)

Datalinks 2 total

Untitled Datalink
via nge0

infiniband-swallower
pkey(8001), Link Mode(cm), via ibp1

Drop the selected device here to create a new datalink, or onto an existing datalink to assign a device.

Interfaces 2 total

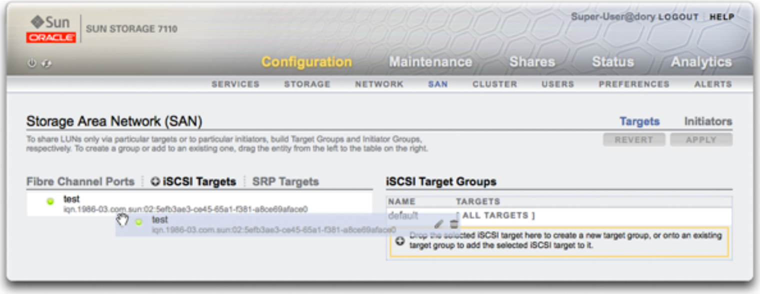
Untitled Interface
IPv4 DHCP, 192.168.2.151/22, via nge0

Untitled Interface
IPv4 static, 10.0.0.1/24, via p8001_ibp1

3. **To create an iSER target, on the Configuration > SAN page, click the iSCSI Targets link.**

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

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



Storage Area Network (SAN)

To share LUNs only via particular targets or to particular initiators, build Target Groups and Initiator Groups, respectively. To create a group or add to an existing one, drag the entity from the left to the table on the right.

test
iqn.1986-03.com.sun:02:5efb3ae3-ce45-65a1-f381-a8ce69aface0

test
iqn.1986-03.com.sun:02:5efb3ae3-ce45-65a1-f381-a8ce69aface0

iSCSI Target Groups

NAME	TARGETS
default	[ALL TARGETS]

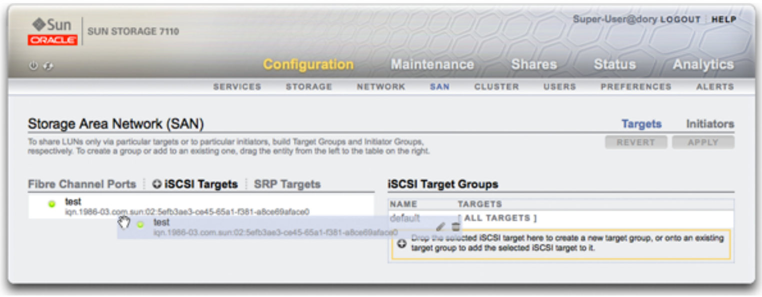
Drop the selected iSCSI target here to create a new target group, or onto an existing target group to add the selected iSCSI target to it.

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

7. **To add a new initiator, click the ⊕ add icon.**

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

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



Storage Area Network (SAN)

To share LUNs only via particular targets or to particular initiators, build Target Groups and Initiator Groups, respectively. To create a group or add to an existing one, drag the entity from the left to the table on the right.

test
iqn.1986-03.com.sun:02:5efb3ae3-ce45-65a1-f381-a8ce69aface0

iSCSI Target Groups

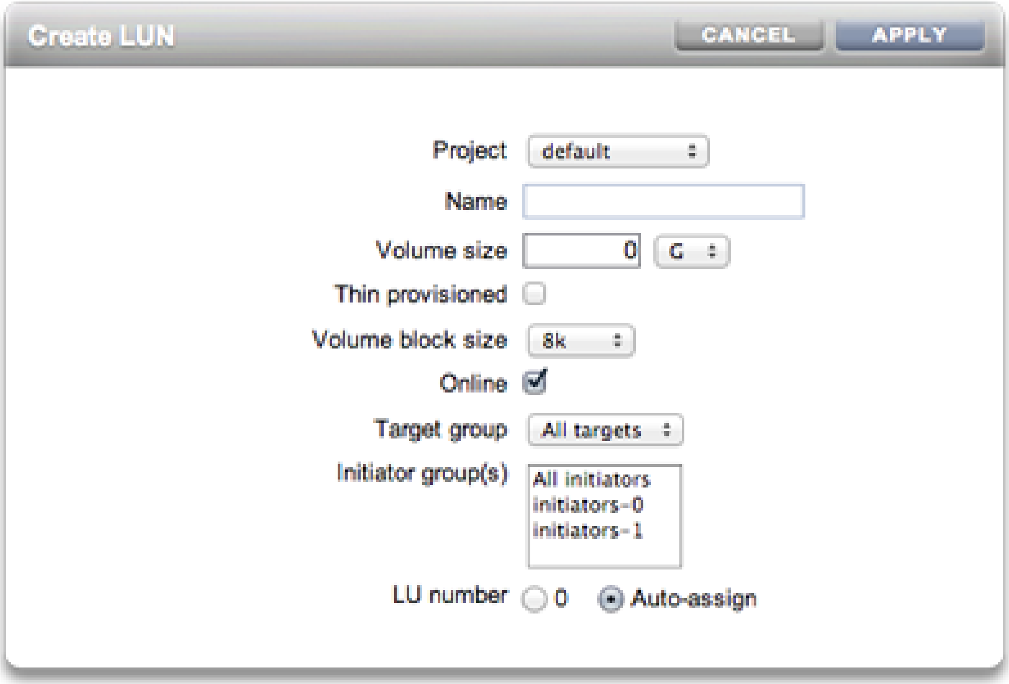
NAME	TARGETS
default	[ALL TARGETS]

Drop the selected iSCSI target here to create a new target group, or onto an existing target group to add the selected iSCSI target to it.

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

11. **Click the ⊕ add icon and associate the new LUN with target or initiator groups you created already using the Target Group and Initiator Groups menu.**

12. **:**



Project: default
Name:
Volume size: 0 G
Thin provisioned: □
Volume block size: 8k
Online: ☑
Target group: All targets
Initiator group(s): All initiators
initiators-0
initiators-1
LU number: ○ 0 ● Auto-assign

Configuring iSCSI Using the CLI

Adding an iSCSI Target with an Auto-generated IQN

```
ahi:configuration san iscsi targets> create
ahi:configuration san iscsi targets target (uncommitted)> set alias="Target 0"
ahi:configuration san iscsi targets target (uncommitted)> set auth=none
ahi:configuration san iscsi targets target (uncommitted)> set interfaces=igb1
ahi:configuration san iscsi targets target (uncommitted)> commit
ahi:configuration san iscsi targets> list
TARGET     ALIAS
target-000 Target 0
           |
           +-> IQN
               iqn.1986-03.com.sun:02:daf0161f-9f5d-e01a-b5c5-e1efa9578416
```

Adding an iSCSI Target with a Specific IQN and RADIUS Authentication

```
ahi:configuration san iscsi targets> create
ahi:configuration san iscsi targets target (uncommitted)> set alias="Target 1"
ahi:configuration san iscsi targets target (uncommitted)>
     set iqn=iqn.2001-02.com.acme:12345
ahi:configuration san iscsi targets target (uncommitted)> set auth=radius
ahi:configuration san iscsi targets target (uncommitted)> set interfaces=igb1
ahi:configuration san iscsi targets target (uncommitted)> commit
ahi:configuration san iscsi targets> list
TARGET     ALIAS
target-000 Target 0
           |
           +-> IQN
               iqn.1986-03.com.sun:02:daf0161f-9f5d-e01a-b5c5-e1efa9578416
target-001 Target 1
           |
           +-> IQN
               iqn.2001-02.com.acme:12345
```

Adding an iSCSI Initiator which uses CHAP Authentication

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

Adding an iSCSI Target Group

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

```
ahi:configuration san iscsi targets groups> list
GROUP     NAME
group-000 tg0
          |
          +-> TARGETS
              iqn.2001-02.com.acme:12345
              iqn.1986-03.com.sun:02:daf0161f-9f5d-e01a-b5c5-e1efa9578416
```

Adding an iSCSI Initiator Group

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

SRP

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

SRP 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.

TABLE 6-5 SRP Target 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:

TABLE 6-6 SRP Target Status Icons

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.

TABLE 6-7 SRP Initiator Properties

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

Observing SRP Performance

SRP performance can be observed via "Analytics" in "Oracle ZFS Storage Appliance Analytics Guide ", whereby one can breakdown operations or throughput by initiator or target. {{Server}}/wiki/images/cfg_san_srp.png

Configuring SRP Targets Using the BUI

▼ SRP Target Configuration

This procedure describes the steps for configuring SRP targets.

1. **Connect HCA ports to IB interfaces.**

2. **: The targets are automatically discovered by the appliance.**
3. **To create the target group, go to the Configuration > SAN screen.**
4. **Click the Target link and then click SRP targets.**
5. **:The SRP targets page appears.**
6. **To create the target group, use the ✥ move icon to drag a target to the Target Groups list.**
7. **Click Apply.**
8. **(Optional) To create an initiator and initiator group on the Initiator screen, click the ⊕ icon, collect GUID from initiator, assign it a name, and drag it to initiator group.**
9. **To create a LUN and associate it with the SRP target and initiators you created in the previous steps, go to the Shares screen.**
10. **Click the LUN link and then click the LUN ⊕ icon. Use the Target Group and Initiator Group menus on the Create LUN dialog to select the SRP groups to associate with the LUN.**

Configuring SRP Targets Using the CLI

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

```
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
```

♦ ♦ ♦ CHAPTER 7

User Configuration

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” on page 236 or “LDAP” on page 238 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.

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

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

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

User 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 “Logs” in “Oracle ZFS Storage Appliance Customer Service Manual ” log.

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

User Authorizations

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

The following table shows the available scopes:

TABLE 7-1 User Available Scopes

Scope BUI	Scope CLI	Example Authorization	Example Filter
Active Directory	ad	Join an Active Directory domain	Domain name
Alerts	alert	Configure alert filters and thresholds	.
Analytics	stat	Read a statistic with this drilldown present	Drilldowns
Clustering	cluster	Failback resources to a cluster peer	.
Datasets	dataset	Manage aspects of Analytics datasets	Configure
Hardware	hardware	Online and offline disks	
Keystores	keystore	Configure keystores.	.
Networking	net	Configure networking devices, datalinks, and interfaces	.
Projects and shares	nas	Change general properties of projects and shares	Pool, project, share
Roles	role	Configure authorizations for a role	Role name
SAN	stmf	Configure authorizations for SAN	
Services	svc	Restart a service	Service name
Shares property schema	schema	Modify property schema	.
System	appliance	Reboot the appliance	Appliance name
Update	update	Update system software	.
Users	user	Change a password	Username

Scope BUI	Scope CLI	Example Authorization	Example Filter
Workflow	workflow	Modify workflow	Workflow name
Worksheet	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.

Managing User Properties

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

User Properties

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

TABLE 7-2 User Properties

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 "Logs" in "Oracle ZFS Storage Appliance Customer Service Manual " 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" on page 48, 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

Property	Description
Roles	The roles possessed by this user
Exceptions	These authorizations are excluded from those normally available due to the selected roles

Role Properties

The following properties can be set when managing roles:

TABLE 7-3 Role Properties

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

Users BUI Page

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

TABLE 7-4 Users BUI Page Icons

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

Configuring Users using the 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.**

▼ 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.**

Configuring Users using the 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.

CLI User Configuration Example

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:igb0   idmap           ndmp            scrk
dns             interface:igb0  nfs             snmp
caji:configuration roles webadmin auth (uncommitted)> set service=http
                       service = http (uncommitted)
caji:configuration roles webadmin auth (uncommitted)> set allow_restart=true
                 allow_restart = true (uncommitted)
caji:configuration roles webadmin auth (uncommitted)> commit
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.

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

6. **Now type `show` to see the full list of preferences. Roles and authorization exceptions may now be added, as well as Chapter 8, "Setting ZFSSA 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 type `commit` to 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. **Type `set` to set the desired authorizations to true, and set the filters (if available). Tab-completion helps show which filter settings are valid.**
8. **Type `commit`. The authorization has now been added.**

▼ 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.**

♦ ♦ ♦ CHAPTER 8

Setting ZFSSA Preferences

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

Preference Properties

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

TABLE 8-1 Preference Settings

Property	Description
Initial login screen	First page the BUI will load after a successful login. By default this is the "Status Dashboard" on page 48.
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" in "Oracle ZFS Storage Appliance Analytics Guide "
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.

Setting Preferences Using the CLI

Preferences can 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
```

Setting SSH Public Keys Using the CLI

SSH Public Keys 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.

♦ ♦ ♦ CHAPTER 9

Alert Configuration

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

Alert Categories

Important appliance events trigger alerts, which includes hardware and software faults. These alerts appear in the “Logs” in “Oracle ZFS Storage Appliance Customer Service Manual ”, and may also be configured to execute any of the Alert actions.

Alerts are grouped into the following categories:

TABLE 9-1 Alert 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 Chapter 11, “ZFSSA Services” failure events

Category	Description
Thresholds	Custom alerts based on “Analytics” in “Oracle ZFS Storage Appliance Analytics Guide ” statistics
ZFS pool	Storage pool events, including scrub and hot space activation

Supported Alert 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” on page 265 service screen.

Send SNMP trap

An SNMP trap containing alert details can be sent, if an SNMP trap destination is configured in the “SNMP” on page 266 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" on page 270 for example syslog payloads and a description of how to configure syslog receivers on other operating systems.

Resume/Suspend Dataset

Analytics "Datasets" in "Oracle ZFS Storage Appliance Analytics Guide " 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 "Open Worksheets" in "Oracle ZFS Storage Appliance Analytics Guide ", which may contain numerous datasets. The reasons for doing this are similar to those for resuming and suspending datasets.

Execute Workflow

Workflows may be optionally executed as alert actions. To allow a workflow to be eligible as an alert action, its `alert` action must be set to `true`. Refer to "Workflows as alert actions" on page 411 for details.

Threshold Alerts

These are alerts based on the statistics from "Analytics" in "Oracle ZFS Storage Appliance Analytics Guide ". The following are properties when creating threshold alerts:

TABLE 9-2 Threshold Alert Properties

Property	Description
Threshold	The threshold statistic is from "Analytics" in "Oracle ZFS Storage Appliance Analytics Guide ", 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
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

Configuring Alerts Using the BUI

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

▼ Adding a Threshold Alert

1. **Click the add icon next to "Threshold alerts".**
2. **Pick the statistic to monitor. You can use “Statistics” in “Oracle ZFS Storage Appliance Analytics Guide ” to view the statistic to check if it is suitable.**
3. **Pick exceeds/falls below, and the desired value.**
4. **Enter the Timing details. The defaults will post the alert only if the threshold has been breached for at least 5 minutes, will repost every 5 minutes, and post after the threshold has cleared for 5 minutes.**
5. **Select the Alert action from the drop down menu, and fill out the required fields on the right.**
6. **If desired, continue to add Alert actions by clicking the add icon next to "Alert actions".**
7. **Click "APPLY" at the top of the dialog.**

▼ Adding an Alert Action

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.**

Configuring Alerts Using the CLI

Alerts can also be configured from the CLI using the `configuration alerts` context. See Tasks for step-by-step instructions for configuring these in the CLI.

▼ Adding a Threshold Alert

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

2. **Enter set statname=where [name is the desired statistic to monitor. To determine the CLI name, enter set statname= and press Tab. For details on each statistic, see “Statistics” in “Oracle ZFS Storage Appliance Analytics Guide ” and click on the statistic names.**

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

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

5. **Enter `list` to determine the name, including number, of the new threshold alert. Look for a threshold with the same limit and statistic name that you just set.**

6. **Enter select threshold-where `[number` is the same number identified in the previous step.**

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

8. **Enter `done`, and then enter `done` again.**

▼ Adding an Alert Action

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

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

3. **Enter `set category=thresholds`.**

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

5. **Enter `commit`.**

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

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

8. **Enter `action`, and then enter `get`.**

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

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

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

12. **Enter `commit`, and then enter `list`. If necessary, correct any arguments now.**

13. **Enter `done`, and then enter `done` again.**

♦ ♦ ♦ CHAPTER 10

Cluster Configuration

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

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

Cluster Features and Benefits

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

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

- Chapter 13, “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” on page 200 LUNs provided by multiple arbitrarily located storage servers,
- Load balancing, which is built into the “NFS” on page 195 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,
- “Problems” in “Oracle ZFS Storage Appliance Customer Service Manual ” 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” on page 257 functionality, and
- Redundant storage devices (RAID).

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

Cluster Disadvantages

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

- The potential for application intolerance of protocol-dependent behaviors during takeover,
- The possibility that the cluster software itself will fail or induce a failure in another subsystem that would not have occurred in standalone operation,
- Increased management complexity and a higher likelihood of operator error when performing management tasks,

- The possibility that multiple failures or a severe operator error will induce data loss or corruption that would not have occurred in a standalone configuration, and
- Increased difficulty of recovering from unanticipated software and/or hardware states.

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

Cluster Terminology

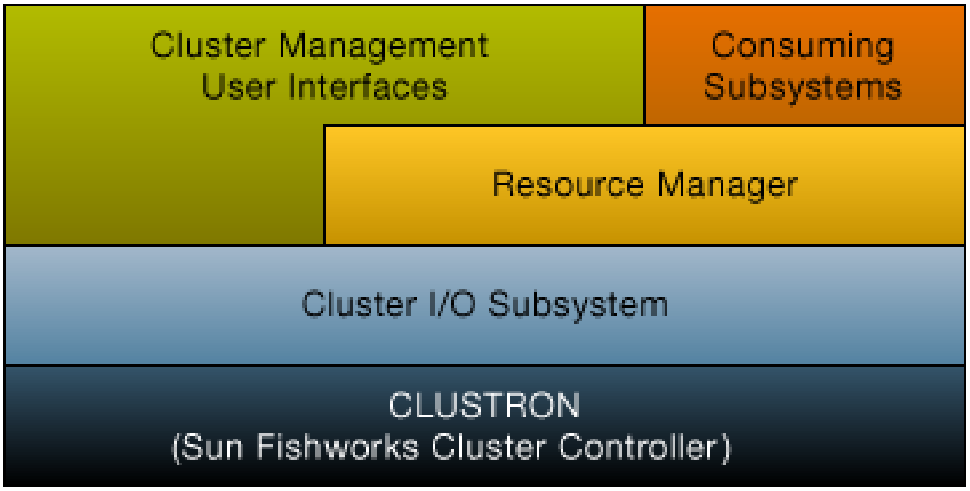
The terms defined here are used throughout the document. In most cases, they are explained in greater context and detail along with the broader concepts involved. The cluster states and resource types are described in the next section. Refer back to this section for reference as needed.

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

Understanding Clustering

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

FIGURE 10-1 Clustering Subsystem



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 Oracle ZFS Storage Appliances are not susceptible to this failure mode. The Ethernet link provides a higher-performance transport for non-heartbeat messages such as rejoin synchronization and provides a backup heartbeat.

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

FIGURE 10-2 ZS3-2 Controller Cluster I/O Ports

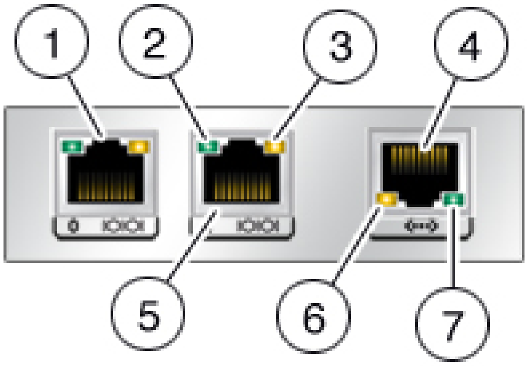


TABLE 10-1 ZS3-2 Controller Cluster I/O Ports

Figure Legend			
1 Serial 0	2 Serial Activity LED	3 Serial Status LED	4 Ethernet
5 Serial 1	6 Ethernet Status LED	7 Ethernet Activity LED	

FIGURE 10-3 ZS3-4 and 7x20 Controller Cluster I/O Ports

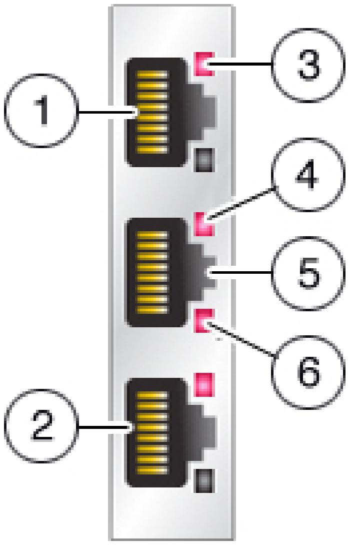


Figure 2. ZS3-4 and 7x20 controller cluster I/O ports

TABLE 10-2 ZS3-4 and 7x20 Controller Cluster I/O Ports

Figure Legend			
1 Serial 1	2 Serial 0	3 Serial Status LED	4 Ethernet Status LED
5 Ethernet	6 Ethernet Activity LED		

Clustered heads only communicate with each other over the secure private network established by the cluster interconnects, and never over network interfaces intended for service or administration. Messages fall into two general categories: regular heartbeats used to detect the failure of a remote head, and higher-level traffic associated with the resource manager and the cluster management subsystem. Heartbeats are sent, and expected, on all three links; they are transmitted continuously at fixed intervals and are never acknowledged or retransmitted as all heartbeats are identical and contain no unique information. Other traffic may be sent over any link, normally the fastest available at the time of transmission, and this traffic is acknowledged, verified, and retransmitted as required to maintain a reliable transport for higher-level software.

Regardless of its type or origin, every message is sent as a single 128-byte packet and contains a data payload of 1 to 68 bytes and a 20-byte verification hash to ensure data integrity. The serial links run at 115200 bps with 9 data bits and a single start and stop bit; the Ethernet link runs at 1Gbps. Therefore the effective message latency on the serial links is approximately 12.2ms. Ethernet latency varies greatly; while typical latencies are on the order of microseconds, effective latencies to the appliance management software can be much higher due to system load.

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

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

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

Note - To avoid data corruption after a physical re-location of a cluster, verify that all cluster cabling is installed correctly in the new location. For more information, see “Preventing 'Split-Brain' Conditions” on page 170

Understanding Cluster Resource Management

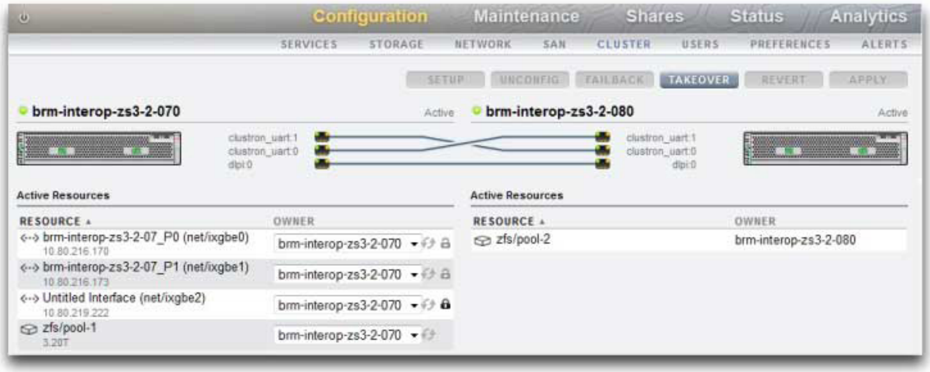
The resource manager is responsible for ensuring that the correct set of network interfaces is plumbed up, the correct storage pools are active, and the numerous configuration parameters remain in sync between two clustered heads. Most of this subsystem's activities are invisible to administrators; however,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

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

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

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

FIGURE 10-4 ZS3-2 Clustering Example



brm-interop-zs3-2-070 Active

Active Resources

RESOURCE ▴	OWNER
brm-interop-zs3-2-07_P0 (net/ixgbe0) 10.80.216.170	brm-interop-zs3-2-070
brm-interop-zs3-2-07_P1 (net/ixgbe1) 10.80.216.173	brm-interop-zs3-2-070
Untitled Interface (net/ixgbe2) 10.80.219.222	brm-interop-zs3-2-070
zfs/pool-1 3.20T	brm-interop-zs3-2-070

brm-interop-zs3-2-080 Active

Active Resources

RESOURCE ▴	OWNER
zfs/pool-2	brm-interop-zs3-2-080

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

TABLE 10-3 Cluster Resource Management

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.

Cluster Takeover and Failback

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

TABLE 10-4 Cluster States

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 takeover). It remains in this state until an administrator manually executes a fail-back operation.
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

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

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

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

Clustering Considerations for Storage

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

Note that in either configuration, any ReadZilla devices can be used only when the pool to which they are assigned is imported on the head that has been assigned ownership of that pool. That is, when a pool has been taken over due to head failure, read caching will not be available for that pool even if the head that has imported it also has unused ReadZillas installed. For this reason, ReadZillas in an active-passive cluster should be configured as described in the Chapter 5, “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.

TABLE 10-5 Clustering Considerations for Storage

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

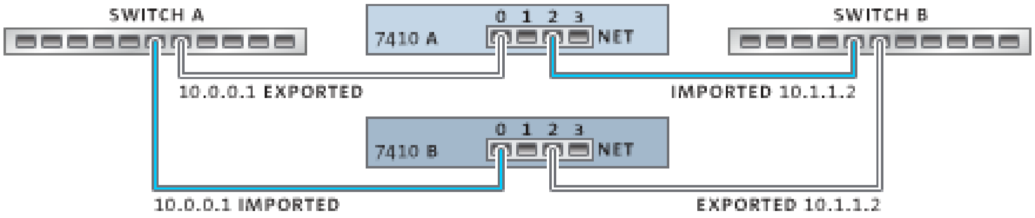
Variable	Single Node ownership	Multiple pools owned by different heads
		by running two workloads on the surviving head instead of the usual one. When nominal workloads on each head approach the head's maximum capabilities, latencies in the failed-over state may be extremely high.
Storage flexibility	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.
Network connectivity	All network devices in each head can be used while that head is providing service.	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 (NSPF). 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 a clustered subsystem head to fail. To protect against network failures inside or outside of the appliance, IPMP and/or LACP should be used. A comprehensive approach to availability requires the correct configuration of the network and a network-wide plan for redundancy.

FIGURE 10-5 Clustering for Networking



Network interfaces can be configured as either singleton or private resources, provided they have a static IP configuration. Interfaces configured using DHCP must be private and using DHCP in clusters is discouraged. When configured as a singleton resource, all datalinks and devices used to construct an interface can be active on only one head at a time. Likewise, corresponding devices on each head must be attached to the same networks in order for service to be provided in a failed-over state. An example of this is shown in the previous diagram.

For a cluster to operate correctly when you construct network interfaces from devices and datalinks, it is essential that each singleton interface has a device using the same identifier and capabilities available on both heads. Since device identifiers depend on the device type and the order in which they are first detected by the appliance, clustered heads MUST have identical hardware installed. Each slot in both heads must be populated with identical hardware and slots must be populated in the same order on both heads. Your qualified Oracle reseller or service representative can assist in planning hardware upgrades that meet these requirements.

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

Example:

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

It is a good idea to assign each clustered head an IP address used only for administration (most likely on a dedicated management network) and to designate the interface as a private resource.

This ensures that it is possible to reach a functioning head from the management network even if it is in a AKCS_STRIPPED state and awaiting failback. This is important if services such as LDAP and Active Directory are in use and require access to other network resources when the head is not providing service. If this is not practical, the service processor should be attached to a reliable network and/or serial terminal concentrator so that the head can be managed using the system console.

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

Private Local IP Interfaces

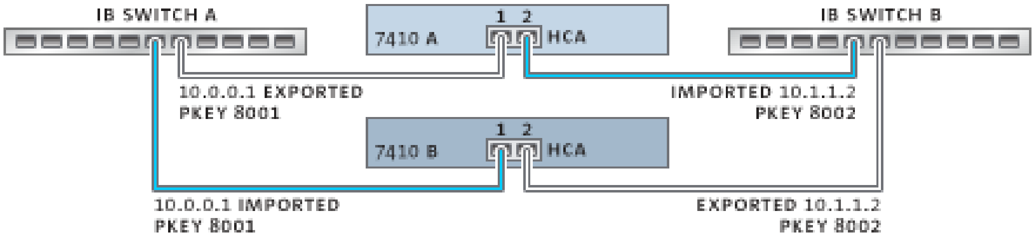
Use the following guidelines when creating private local IP interfaces:

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

Clustering Considerations for Infiniband

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

FIGURE 10-6 Clustering Considerations for Infiniband

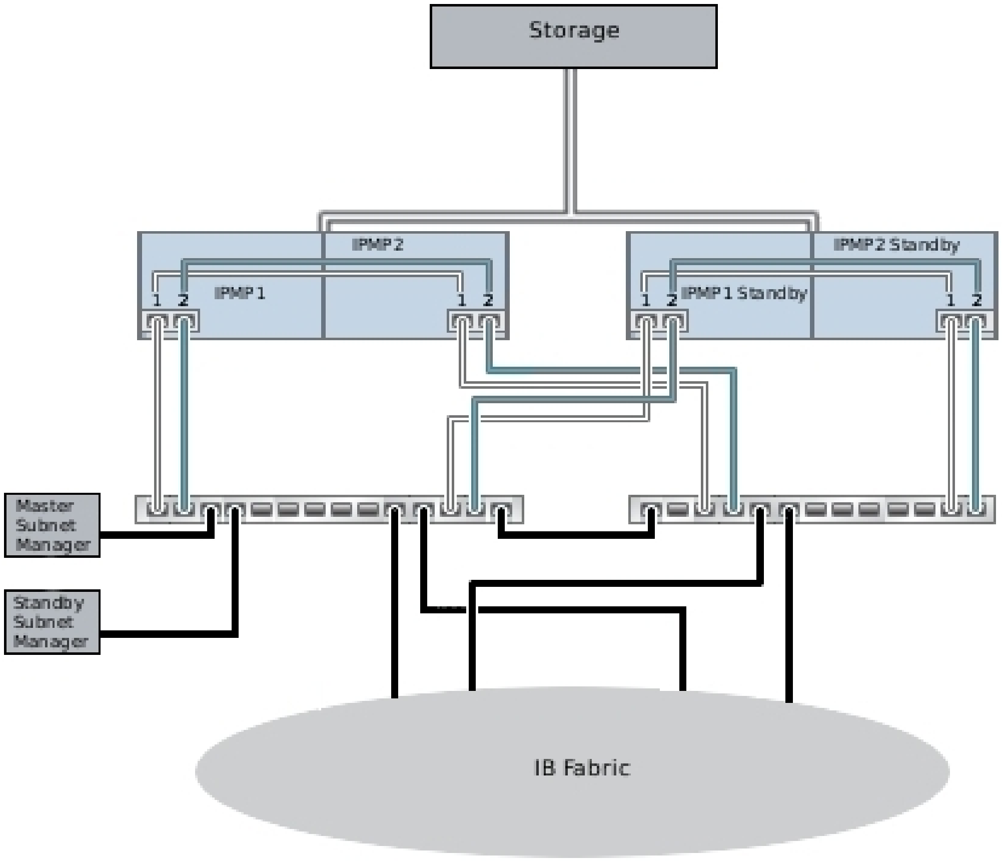


To ensure proper cluster configuration, each head must be populated with identical HCAs in identical slots. Furthermore, each corresponding HCA port must be configured into the same partition (pkey) on the subnet manager with identical membership privileges and attached to the same network. To reduce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

Clustering Redundant Path Scenarios

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

FIGURE 10-7 Cluster Configuration for Subnet Manager Redundancy



Preventing 'Split-Brain' Conditions

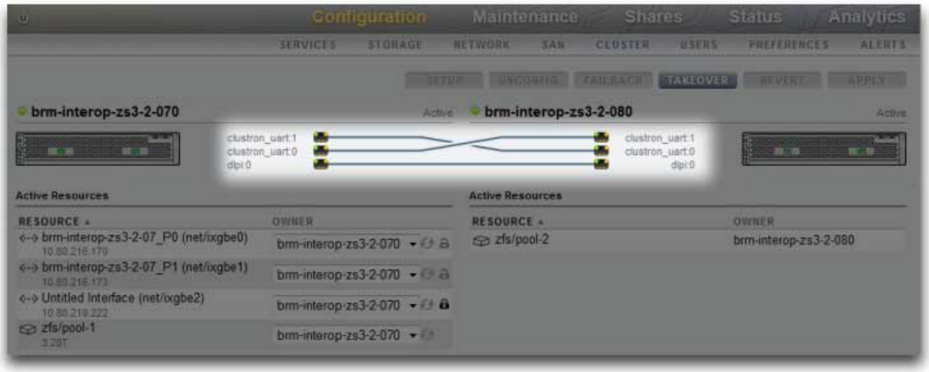
A common failure mode in clustered systems is known as *split-brain*; in this condition, each of the clustered heads believes its peer has failed and attempts takeover. Absent additional logic, this condition can cause a broad spectrum of unexpected and destructive behavior that can be difficult to diagnose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

A number of arbitration mechanisms are employed by similar products; typically they entail the use of *quorum disks* (using SCSI reservations) or *quorum servers*. To support the use of ATA disks without the need for additional hardware, the Oracle ZFS Storage Appliance uses a different approach relying on the storage fabric itself to provide the required mutual exclusivity. The arbitration process consists of attempting to perform a SAS ZONE LOCK command on each of the visible SAS expanders in the storage fabric, in a predefined order. Whichever appliance is successful in its attempts to obtain all such locks will proceed with takeover; the other will reset itself. Since a clustered appliance that boots and detects that its peer is unreachable will attempt takeover and enter the same arbitration process, it will reset in a continuous loop until at least one cluster I/O link is restored. This ensures that the subsequent failure of the other head will not result in an extended outage. These SAS zone locks are released when failback is performed or approximately 10 seconds has elapsed since the head in the AKCS_OWNER state most recently renewed its own access to the storage fabric.

This arbitration mechanism is simple, inexpensive, and requires no additional hardware, but it relies on the clustered appliances both having access to at least one common SAS expander in the storage fabric. Under normal conditions, each appliance has access to all expanders, and arbitration will consist of taking at least two SAS zone locks. It is possible, however, to construct multiple-failure scenarios in which the appliances do not have access to any common expander. For example, if two of the SAS cables are removed or a 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 Oracle ZFS Storage Appliance is designed to ensure that there is no single point of failure, and to protect both data and availability against any plausible failure without adding undue cost or complexity to the system. It is still possible that massive multiple failures will cause loss of service and/or data, in much the same way that no RAID layout can protect against an unlimited number of disk failures.

FIGURE 10-8 Preventing Split-Brain



Active Resources

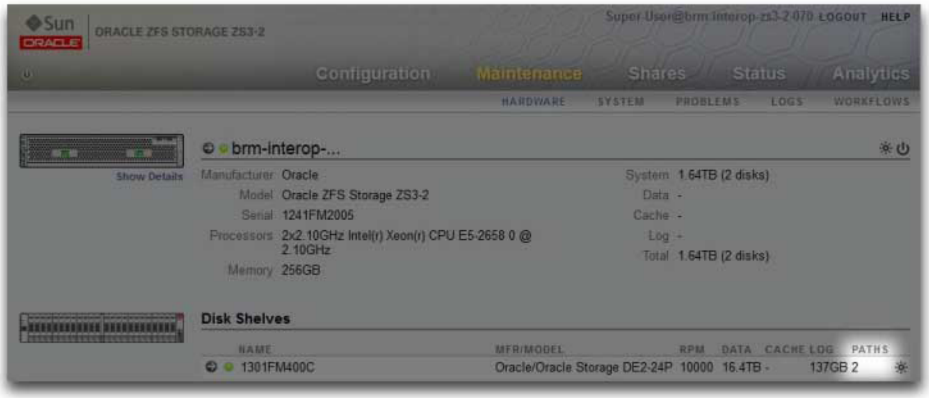
RESOURCE ▲	OWNER
brm-interop-zs3-2-07_P0 (net/ixgbe0) 10.80.216.170	brm-interop-zs3-2-070
brm-interop-zs3-2-07_P1 (net/ixgbe1) 10.80.216.173	brm-interop-zs3-2-070
Untitled Interface (net/ixgbe2) 10.80.219.222	brm-interop-zs3-2-070
zfs/pool-1 3.20T	brm-interop-zs3-2-070

Active Resources

RESOURCE ▲	OWNER
zfs/pool-2	brm-interop-zs3-2-080

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.

FIGURE 10-9 Cluster Two Paths



brm-interop-...

Manufacturer Oracle
Model Oracle ZFS Storage ZS3-2
Serial 1241FM2005
Processors 2x2.10GHz Intel(r) Xeon(r) CPU E5-2658 0 @ 2.10GHz
Memory 256GB

System 1.64TB (2 disks)
Data -
Cache -
Log -
Total 1.64TB (2 disks)

Disk Shelves

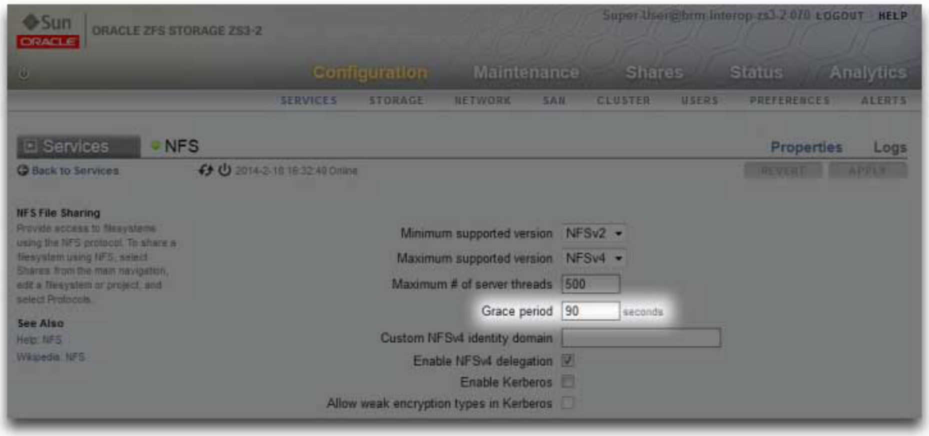
NAME	MFR/MODEL	RPM	DATA	CACHE	LOG	PATHS
1301FM400C	Oracle/Oracle Storage DE2-24P	10000	16.4TB	-	137GB	2

Estimating and Reducing Takeover Impact

There is an interval during takeover and failback during which access to storage cannot be provided to clients. The length of this interval varies by configuration, and the exact effects on clients depends on the protocol(s) they are using to access data. Understanding and mitigating these effects can make the difference between a successful cluster deployment and a costly failure at the worst possible time.

NFS (all versions) clients typically hide outages from application software, causing I/O operations to be delayed while a server is unavailable. NFSv2 and NFSv3 are stateless protocols that recover almost immediately upon service restoration. NFSv4 incorporates a client grace period at startup, during which I/O typically cannot be performed. The duration of this grace period can be tuned in the Oracle ZFS Storage Appliance (see illustration); reducing it will reduce the apparent impact of takeover and/or failback. For planned outages, the Oracle ZFS Storage Appliance provides grace-less recovery for NFSv4 clients, which avoids the grace period delay. For more information about grace-less recovery, see the Grace period property in NFS “Properties” on page 195.

FIGURE 10-10 Cluster Grace Period



NFS File Sharing

Provide access to filesystems using the NFS protocol. To share a filesystem using NFS, select Shares from the main navigation, edit a filesystem or project, and select Protocols.

See Also

Help: NFS

Wikipedia: NFS

Minimum supported version NFSv2

Maximum supported version NFSv4

Maximum # of server threads 500

Grace period 90 seconds

Custom NFSv4 identity domain

Enable NFSv4 delegation ☑

Enable Kerberos ☐

Allow weak encryption types in Kerberos ☐

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 - on the order of tens of milliseconds - but with very large share counts this can contribute significantly to takeover times. Keeping the number of shares relatively small - a few thousand or fewer - can therefore reduce these times considerably.

Failback time is normally greater than takeover time for any given configuration. This is because failback is a two-step operation: first, the source appliance exports all resources of which it is not the assigned owner, then the target appliance performs the standard takeover procedure on its own assigned resources only. Therefore it will always take longer to failback from head A to head B than it will take for head A to take over from head B in case of failure. This additional failback time is much less dependent upon the number of disksets being exported than is the takeover time, so keeping the number of shares and LUNs small can have a greater impact on failback than on takeover. It is also important to keep in mind that failback is always initiated by an administrator, so the longer service interruption it causes can be scheduled for a time when it will cause the lowest level of business disruption.

Note: Estimated times cited in this section refer to software/firmware version 2009.04.10,1-0. Other versions may perform differently, and actual performance may vary. It is important to test takeover and its exact impact on client applications prior to deploying a clustered appliance in a production environment.

Cluster Configuration Using the BUI

To configure or unconfigure a cluster, use the following procedures.

Unconfiguring clustering is a destructive operation that returns one of the clustered storage controllers to its factory default configuration and reassigns ownership of all resources to the surviving peer. There are two reasons to unconfiguring clustering. You no longer wish to use clustering; instead, you wish to configure two independent storage appliances. You are replacing a failed storage controller with new hardware or a storage controller with factory-fresh appliance software (typically this replacement is performed by your service provider).

▼ Configuring Clustering

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 appliance.**

4. **Power on both appliances - but do not begin configuration. Select only one of the two appliances from which you will perform configuration; the choice is arbitrary. This will be referred to as the primary appliance for configuration purposes. Connect to and access the serial console of that appliance, and perform the initial tty-based configuration on it in the same manner as you would when configuring a standalone appliance. Note: Do not perform the initial tty-based configuration on the secondary appliance; it will be automatically configured for you during cluster setup.**

5. **On the primary appliance, enter either the BUI or CLI to begin cluster setup. Cluster setup can be selected as part of initial setup if the cluster interconnect controller has been installed. Alternately, you can perform standalone configuration at this time, deferring cluster setup until later. In the latter case, you can perform the cluster configuration task by clicking the Setup button in Configuration->Cluster.**

6. **At the first step of cluster setup, you will be shown a diagram of the active cluster links: you should see three solid blue wires on the screen, one for each connection. If you don't, add the missing cables now. Once you see all three wires, you are ready to proceed by clicking the Commit button.**

7. **Enter the appliance name and initial root password for the second appliance (this is equivalent to performing the initial serial console setup for the new appliance). When you click the Commit button, progress bars will appear as the second appliance is configured.**

8. **If you are setting up clustering as part of initial setup of the primary appliance, you will now be prompted to perform initial configuration as you would be in the single-appliance case. All configuration changes you make will be propagated automatically to the other appliance. Proceed with initial configuration, taking into consideration the following restrictions and caveats: Network interfaces configured via DHCP cannot be failed over between heads, and therefore cannot be used by clients to access storage. Therefore, be sure to assign static IP addresses to any network interfaces which will be used by clients to access storage. If you selected a DHCP-configured network interface during tty-based initial configuration, and you wish to use that interface for client access, you will need to change its address type to Static before proceeding. Best practices include configuring and assigning a private network interface for administration to each head, which will enable administration via either head over the network (BUI or CLI) regardless of the cluster state. If routes are needed, be sure to create a route on an interface that will be assigned to each head. See the previous section for a specific example.**

9. **Proceed with initial configuration until you reach the storage pool step. Each storage pool can be taken over, along with the network interfaces clients use to reach that storage pool, by the cluster peer when takeover occurs. If you**

create two storage pools, each head will normally provide clients with access to the pool assigned to it; if one of the heads fails, the other will provide clients with access to both pools. If you create a single pool, the head which is not assigned a pool will provide service to clients only when its peer has failed. Storage pools are assigned to heads at the time you create them; the storage configuration dialog offers the option of creating a pool assigned to each head independently. The smallest unit of storage that may be assigned to a pool is one disk. If you create multiple pools, there is no requirement that they must be the same size. Note that fewer pools with more disks per pool are preferred because they simplify management and provide a higher percentage of overall usable capacity. It is recommended that each pool includes a minimum of 8 disks, and ideally more, across all JBODs.

10. **After completing basic configuration, you will have an opportunity to assign resources to each head. Typically, you will need to assign only network interfaces; storage pools were automatically assigned during the storage configuration step.**

11. **Commit the resource assignments and perform the initial fail-back from the Cluster User Interface, described below. If you are still executing initial setup of the primary appliance, this screen will appear as the last in the setup sequence. If you are executing cluster setup manually after an initial setup, go to the Configuration/Cluster screen to perform these tasks. Refer to Cluster User Interface below for the details.**

▼ Unconfiguring Clustering

1. **Select the storage controller that will be reset to its factory configuration. Note that if replacing a failed storage controller, you can skip to step 3, provided that the failed storage controller will not be returned to service at your site.**

2. **From the system console of the storage controller that will be reset to its factory configuration, perform a factory reset.**

3. **The storage controller will reset, and its peer will begin takeover normally. NOTE: Prior to allowing the factory-reset storage controller to begin booting (i.e., prior to progressing beyond the boot menu), power it off and wait for its peer to complete takeover.**

4. **Detach the cluster interconnect cables (see above) and detach the powered-off storage controller from the cluster's external storage enclosures.**

5. **On the remaining storage controller, click the Unconfig button on the Configuration -> Clustering screen. All resources will become assigned to that**

storage controller, and the storage controller will no longer be a member of any cluster.

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

Note - If your cluster had 2 or more pools, ownership of all pools will be assigned to the remaining storage controller after unconfiguration. In software versions prior to 2010.Q1.0.0, this was not a supported configuration; if you are running an older software version, you must do one of: destroy one or both pools, attach a replacement storage controller, perform the cluster setup task described above, and reassign ownership of one of the pools to the replacement storage controller, or upgrade to 2010.Q1.0.0 or a later software release which contains support for multiple pools per storage controller.

Configuring Clustering Using the CLI

▼ Shutting Down a Clustered Configuration

1. **Verify the cluster state, using the following CLI commands:**

```
nas-7420-1a:> configuration cluster
nas-7420-1a:configuration cluster> show
```

2. **The following is an example of the cluster properties: state indicates the status of the head where you ran the command; peer_state indicates the status of the other head.**

```
state = AKCS_OWNER
description = Active (takeover completed)
peer_asn = 365ed33c-3b9d-c533-9349-8014e9da0408
peer_hostname = nas-7420-1b
peer_state = AKCS_STRIPPED
peer_description = Ready (waiting for failback)
```

3. **Use the following table to verify the node status.**

This Node	Other Node	Condition
AKCS_CLUSTERED	AKCS_CLUSTERED	Both nodes are running in normal condition.
AKCS_OWNER	AKCS_STRIPPED	This node has all the resources and is in active node. The other node is in stand-by and has no resources.
AKCS_OWNER	rebooting	Another node is rebooting and this node has all resources.
AKCS_OWNER	unknown	This node does not know the partner.

Note - If the status of the heads DOES NOT agree, the cluster may be experiencing a problem. Contact Oracle Support before proceeding.

▼ Shutdown the Stand-by Head

1. **Shutdown the stand-by head, using the CLI to run the following commands:**

```
nas-7420-1b:configuration cluster> cd /
nas-7420-1b:> maintenance system poweroff
This will turn off power to the appliance. Are you sure? (Y/N)
```

2. **To verify that you want to shut down the other head, type Y.**

Note - If both heads have a status of AKCS_CLUSTERED, a takeover of the surviving head begins automatically.

3. **Confirm that the stand-by head is powered off, and the cluster state is OWNER/ unknown.**

4. **Shut down the active head, using the CLI to run the following commands:**

```
nas-7420-1a:configuration cluster> cd /
nas-7420-1a:> maintenance system poweroff
This will turn off power to the appliance. Are you sure? (Y/N)
```

5. **To verify that you want to shut down the active head, type Y.**

6. **Confirm that both heads are powered off. From the ILOM prompt, run:**

   ```
   -> show /SYS power_state
   ```

7. **Power off the disk shelves.**

Unconfiguring Clustering

- **Unconfiguring clustering in the CLI operates the same as the BUI unconfig button. If a user attempts to unconfig a cluster when it is not in a correct state, an error appears.**

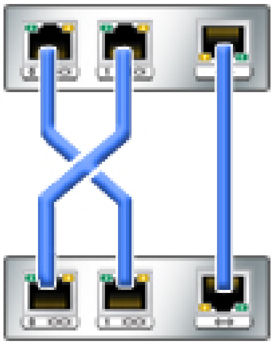
```
configuration cluster> help
Subcommands that are valid in this context:

   resources            => Configure resources

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

   show                 => Show information pertinent to the current context

   done                 => Finish operating on "cluster"

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

   setup                => Run through initial cluster setup

   failback             => Fail back all resources assigned to the cluster peer

   takeover             => Take over all resources assigned to the cluster peer

   unconfig             => Unconfigure the cluster

   links                => Report the state of the cluster links
```

Cluster Node Cabling

Clustered head nodes must be connected together using the cluster interconnect ports located at the back of the controller.

ZS3-2 Cluster Cabling

FIGURE 10-11 ZS3-2 Cluster Cabling



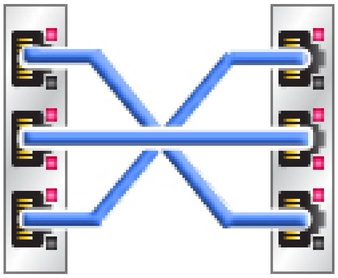
The ZS3-2 controller provides three redundant links that enable the heads to communicate: two serial links (the first two connectors) and an Ethernet link (the third connector).

Using straight-through Cat 5-or-better Ethernet cables, (three 1m cables ship with your cluster configuration), connect the head node according to the diagram at left.

The cluster cabling can be performed either prior to powering on either head node, or can be performed live while executing the cluster setup guided task. The user interface will show the status of each link, as shown later in this page. You must have established all three links before cluster configuration will proceed.

ZS3-4 and 7x20 Cluster Cabling

FIGURE 10-12 ZS3-4 and 7x20 Cluster Cabling



The ZS3-4 and 7x20 controllers provide three redundant links that enable the heads to communicate: two serial links (the outer two connectors) and an Ethernet link (the middle connector).

Using straight-through Cat 5-or-better Ethernet cables, (three 1m cables ship with your cluster configuration), connect the head node according to the diagram at left.

The cluster cabling can be performed either prior to powering on either head node, or can be performed live while executing the cluster setup guided task. The user interface will show the status of each link, as shown later in this page. You must have established all three links before cluster configuration will proceed.

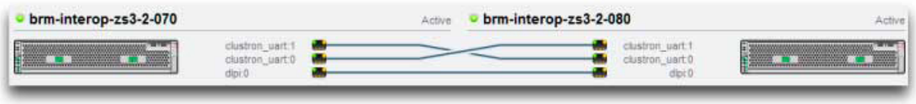
Storage Shelf Cabling

You need to attach your storage shelves to both appliances before beginning cluster configuration. See “Installation” in “Oracle ZFS Storage Appliance Installation Guide ” or follow the Quick Setup poster that shipped with your system.

Cluster Configuration BUI Page

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.

FIGURE 10-13 Configuration Cluster View



The interface contains the following objects:

- A thumbnail picture of each system, with the system whose administrative interface is being accessed shown at left. Each thumbnail is labeled with the canonical appliance name, and its current cluster state (the icon above, and a descriptive label).
- A thumbnail of each cluster card connection that dynamically updates with the hardware: a solid line connects a link when that link is connected and active, and the line disappears if that connection is broken or while the other system is restarting/rebooting.
- A list of the PRIVATE and SINGLETON resources (see Introduction, above) currently assigned to each system, shown in lists below the thumbnail of each cluster node, along with various attributes of the resources.
- For each resource, the appliance to which that resource is assigned (that is, the appliance that will provide the resource when both are in the CLUSTERED state). When the current appliance is in the OWNER state, the owner field is shown as a pop-up menu that can be edited and then committed by clicking Apply.
- For each resource, a lock icon indicating whether or not the resource is PRIVATE. When the current appliance is in either of the OWNER or CLUSTERED states, a resource can be locked to it (made PRIVATE) or unlocked (made a SINGLETON) by clicking the lock icon and then clicking Apply. Note that PRIVATE resources belonging to the remote peer will not be displayed on either resource list.

The BUI contains the following buttons:

TABLE 10-6 Shelf Cabling Interface 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.

Button	Description
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

♦ ♦ ♦ CHAPTER 11

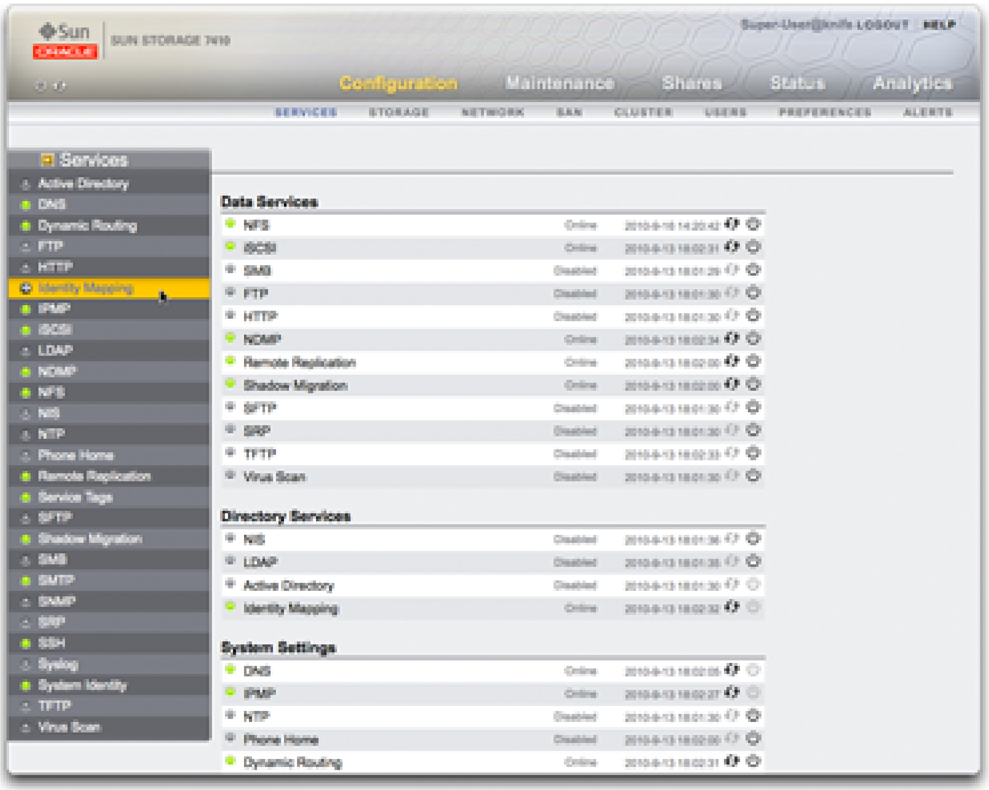
ZFSSA Services

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

Available Services

You can configure the following ZFSSA services:

FIGURE 11-1 Services Configuration BUI Page



Data Services

NFS	Online	2010-8-16 14:20:42
iSCSI	Online	2010-8-13 18:02:31
SMB	Disabled	2010-8-13 18:01:29
FTP	Disabled	2010-8-13 18:01:30
HTTP	Disabled	2010-8-13 18:01:30
NDMP	Online	2010-8-13 18:02:34
Remote Replication	Online	2010-8-13 18:02:00
Shadow Migration	Online	2010-8-13 18:02:00
SFTP	Disabled	2010-8-13 18:01:30
SRP	Disabled	2010-8-13 18:01:30
TFTP	Disabled	2010-8-13 18:02:33
Virus Scan	Disabled	2010-8-13 18:01:30

Directory Services

NIS	Disabled	2010-8-13 18:01:36
LDAP	Disabled	2010-8-13 18:01:35
Active Directory	Disabled	2010-8-13 18:01:30
Identity Mapping	Online	2010-8-13 18:02:32

System Settings

DNS	Online	2010-8-13 18:02:05
IPMP	Online	2010-8-13 18:02:27
NTP	Disabled	2010-8-13 18:01:30
Phone Home	Disabled	2010-8-13 18:02:00
Dynamic Routing	Online	2010-8-13 18:02:31

Data Services

TABLE 11-1 Available Data Services

Service	Description	Ports Used
"NFS" on page 195	Filesystem access via the NFSv3 and NFSv4 protocols	111 and 2049
"iSCSI" on page 200	LUN access via the iSCSI protocol	3260 and 3205
"SMB" on page 202	Filesystem access via the SMB protocol	SMB-over-NetBIOS 139
SMB-over-TCP 445		

Service	Description	Ports Used
NetBIOS Datagram 138		
NetBIOS Name Service 137		
"FTP" on page 217	Filesystem access via the FTP protocol	21
"HTTP" on page 219	Filesystem access via the HTTP protocol	80
"NDMP" on page 221	NDMP host service	10000
"Remote Replication" on page 228	Remote replication	216
"Shadow Migration" on page 229	Shadow data migration	
"SFTP" on page 229	Filesystem access via the SFTP protocol	218
"SRP" on page 232	Block access via the SRP protocol	
"TFTP" on page 233	Filesystem access via the TFTP protocol	
"Virus Scan" on page 233	Filesystem virus scanning	

Directory Services

Note: UIDs and GIDs from 0-99 are reserved by the operating system vendor for use in future applications. Their use by end system users or vendors of layered products is not supported and may cause security related issues with future applications.

TABLE 11-2 Available Directory Services

Service	Description	Ports Used
"NIS" on page 236	Authenticate users and groups from an NIS service	
"LDAP" on page 238	Authenticate users and groups from an LDAP directory	389
"Active Directory" on page 242	Authenticate users with a Microsoft Active Directory Server	
"Identity Mapping" on page 247	Map between Windows entities and Unix IDs	

Service Settings

TABLE 11-3 Service Settings

Service	Description	Ports Used
“DNS” on page 254	Domain name service client	53
“Dynamic Routing” on page 256	RIP and RIPng dynamic routing protocols	
“IPMP” on page 257	IP Multipathing for IP fail-over	
“NTP” on page 258	Network time protocol client	
“Phone Home” on page 261	Product registration and support configuration	443
“Service Tags” on page 265	Product inventory support	443
“SMTP” on page 265	Configure outgoing mail server	
“SNMP” on page 266	SNMP for sending traps on alerts and serving appliance status information	
“Syslog” on page 270	Syslog Relay for sending syslog messages on alerts and forwarding service syslog messages	
“System Identity” on page 275	System name and location	

Remote Access Services

TABLE 11-4 Available Remote Access Services

Service	Description	Ports Used
“SSH” on page 276	SSH for CLI access	22
“REST” on page 264	RESTful API	

Security Services

TABLE 11-5 Available Security Services

Service	Description	Ports Used
Kerberos	Kerberos V Authentication	88

Service	Description	Ports Used
Kerberos V Change & Set Password (SET_CHANGE)	464	
Kerberos V Change & Set Password (RPCSEC_GSS)	749	

Minimum Needed Ports

To provide security on a network, you can deploy firewalls within your network architecture. Port numbers are used for creating firewall rules and uniquely identify a transaction over a network by specifying the host and the service.

The following list shows the minimum ports required for creating firewall rules that allow full functionality of the appliance:

Inbound Ports

- icmp/0-65535 (PING)
- tcp/1920 (EM)
- tcp/215 (BUI)
- tcp/22 (SSH)
- udp/161 (SNMP)

Outbound Ports

- tcp/80 (WEB)
- tcp/443 (SSL WEB)

Note - An outbound port of tcp/443 is used for sending Phone Home messages, uploading support bundles, and update notifications. For replication, use Generic Routing Encapsulation (GRE) tunnels when possible. This lets traffic run on the back end interfaces and avoid the firewall where traffic could be slowed. If GRE tunnels are not available on the NFS core, you must run replication over the front end interface. In this case, port 216 must also be open.

Configuring Services Using the BUI

You use the BUI Services screens to view and modify the services and settings described in tables above. Double click a service line to view the definition screen for that service. The following tables describes the icons and buttons in the services screens:

TABLE 11-6 Services BUI Page Icons and Buttons

Icon	Description
➔	Go to the service screen to configure properties and view logs. This button appears when you mouse-over a service
🟢	The service is enabled and working normally.
⚪	The service is offline or disabled.
🟠	The service has a problem and requires operator attention.
⏻	Enables or disables the service
⟳	Restarts the service
⏻	Enable/disable not available for this service
⟳	Restarts the currently unavailable service. You must enable the service first)

▼ Viewing a Specific Service Screen

1. **To view or edit the properties for a specific service, mouse over the service the status icon that is to the left of the service name.**

2. **The status icon turns into an arrow icon, which you click to display the properties screen for the selected service.**

▼ Viewing a Specific Service Screen

- **In any of the services screens, you can show a side panel of all services by clicking the small arrow icon to the left of the Services title (near the top left of each screen). Click this icon again to hide the list.**

▼ Enabling a Service

- **If a service is not online, click the power icon ⏻ to bring the service online 🟢**

▼ Disabling a Service

- **If a service is online and you want to disabled it, click the power icon ⏻ to take the service offline ●**

▼ Defining Properties

1. **To define properties for a service, double click a service.**
2. **Change the properties and then click APPLY.**
3. **To reset properties, click REVERT.**

▼ Viewing Service Logs

1. **Some services provide service logs with information to help you diagnose service issues. If a Logs button exists in the top right of a service screen, that service provides a log. Logs can provide the following information:**
   - Times when a service changed state
   - Error messages from the service
2. **Log content is specific to each individual service and is subject to change with future updates to the appliance software. The following are example messages that are commonly used in this version of the appliance:**

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

Configuring Services Using the CLI

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

The following example is from the "NTP" on page 258 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 first log event in the example shows that the system was booted at 21:05. The second entry at 21:07:37 records that the service began startup, which completed at 21:13:38. Due to the nature of NTP and system clock adjustment, this service can take minutes to complete startup, as shown by the log.

```
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

1. **After you select a service, you can view its state, enable it, disable it, and set its properties.**

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

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

Viewing a Service's State

- **You can view a service's state using the `show` command:**

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

Enabling a Service

- **Use the `enable` command to enable a service:**

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

Disabling a Service

- **Use the `disable` command to disable a service:**

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

Setting Properties

1. **Use the `set` command to set the properties for the selected service.**

2. **After setting the properties, use the `commit` command to save and activate the new 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 =
```

3. **Property names are similar to their names in the BUI, but CLI names are usually shorter and sometimes abbreviated.**

▼ Viewing 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

Network File System (NFS) is an industry standard protocol to share files over a network. The Sun ZFS Storage Appliance supports NFS versions 2, 3, and 4. For more information on how the filesystem namespace is constructed, see the “filesystem namespace” on page 291 section. For information about NFS with local users, see Chapter 7, “User Configuration”.

Properties

- Minimum supported version - Use this drop-down list to control which versions of NFS the appliance supports.

- Maximum supported version - Use this drop-down list to control which versions of NFS the appliance supports.
- Maximum # of server threads - Define the maximum number of concurrent NFS requests (from 20 to 1000). This should at least cover the number of concurrent NFS clients that you anticipate.
- Grace period - Define the number of seconds that all clients have to recover locking state after an appliance reboot (from 15 to 600 seconds) from an unplanned outage. This property affects only NFS v4 clients (NFS v3 is stateless so there is no state to reclaim). During this period, the NFS service only processes reclaims of the old locking state. No other requests for service are processed until the grace period is over. The default grace period is 90 seconds. Reducing the grace period lets NFS clients resume operation more quickly after a server reboot, but increases the probability that a client cannot recover all of its locking state. The Oracle ZFS Storage Appliance provides grace-less recovery of the locking state for NFSv4 clients during planned outages. Planned outages occur during events such as "Updates" in "Oracle ZFS Storage Appliance Customer Service Manual ", and appliance reboot using the CLI `maintenance system reboot` command, or rebooting using the BUI power icon ⏻. For planned outages, the NFS service processes all requests for service without incurring the grace period delay.
- Custom NFSv4 identity domain - Use this property to define the domain for mapping NFSv4 users and group identities. If you do not set this property, the appliances uses DNS to obtain the identity domain, first by checking for a `_nfsv4idmapdomain` DNS resource record, and then by falling back to the DNS domain itself.
- Enable NFSv4 delegation - Select this property to allow clients to cache files locally and make modifications without contacting the server. This option is enabled by default and typically results in better performance; but in rare circumstances it can cause problems. You should only disable this setting after careful performance measurements of your particular workload and after validating that the setting has a measurable performance benefit. This option only affects NFSv4 mounts.
- Mount visibility - This property lets you limit the availability of information about share access lists and remote mounts from NFS clients. Full allows full access. Restricted restricts access such that a client can see only the shares which it is allowed to access. A client cannot see access lists for shares defined at the server or remote mounts from the server done by other clients. The property is set to Full by default.
- Enable Kerberos - Enables/disables Kerberos service.
- * Allow weak encryption types in Kerberos - Enables/disables support for DES (des-cbc-crc, des-cbc-md5) and Exportable ArcFour with HMAC/md5 (arcfour-hmac-exp). This property is disabled by default.
- * Kerberos realm - A realm is logical network, similar to a domain, that defines a group of systems that are under the same master KDC. Realm names can consist of any ASCII string. Usually, your realm name is the same as your DNS domain name, except that the realm name is in uppercase. Using this convention helps you differentiate problems with the Kerberos service from problems with the DNS namespace, while still using a name that is familiar.

- * Kerberos master KDC - In each realm, you must include a server that maintains the master copy of the principal database. The most significant difference between a master KDC and a slave KDC is that only the master KDC handles database administration requests. For instance, you must change a password or add a new principal on the master KDC.
- * Kerberos slave KDC - The slave contains duplicate copies of the principal database. Both the master KDC server and the slave KDC server create tickets that are used to establish authentication.
- * Kerberos admin principal - This property identifies the administrator. By convention, a principal name is divided into three components: the primary, the instance, and the realm. You can specify a principal as `joe`, `joe/admin`, or `joe/admin@ENG.EXAMPLE.COM`. This property is used only to set up the system's Kerberos service principals and is not retained.
- * Kerberos admin password - Defines a password for the administrator. This property is used only to set up the system's Kerberos service principals and is not retained.
- Oracle Intelligent Storage Protocol - The NFSv4 service includes support for the Oracle Intelligent Storage Protocol, which lets Oracle Database NFSv4 clients pass optimization information to the ZFS Storage Appliance NFSv4 server. For more information, see “ Oracle Intelligent Storage Protocol ” on page 463.

Changing services properties is documented in the “Configuring Services Using the BUI” on page 189 and “Configuring Services Using the CLI” on page 192.

Setting the NFS minimum and maximum versions to the same value causes the appliance to only communicate with clients using that version. This may be useful if you find an issue with one NFS version or the other (such as the performance characteristics of an NFS version with your workload), and you want to force clients to only use the version that works best.

Kerberos Realms

Configuring a Kerberos realm creates certain service principals and adds the necessary keys to the system's local keytab. The “NTP service” on page 258 must be configured before configuring Kerberized NFS. The following service principals are created and updated to support Kerberized NFS:

```
host/node1.example.com@EXAMPLE.COM
nfs/node1.example.com@EXAMPLE.COM
```

If you clustered your appliances, principals and keys are generated for each cluster node:

```
host/node1.example.com@EXAMPLE.COM
nfs/node1.example.com@EXAMPLE.COM
host/node2.example.com@EXAMPLE.COM
nfs/node2.example.com@EXAMPLE.COM
```

If these principals have already been created, configuring the realm resets the password for each of those principals. If you configured your appliance to join an Active Directory domain, you cannot configure it to be part of a Kerberos realm.

For information on setting up KDCs and Kerberized clients, see `http://docs.oracle.com/cd/E26502_01/html/E29015/index.html. (http://docs.oracle.com/cd/E26502_01/html/E29015/index.html.)` After setting NFS properties for Kerberos, change the Security mode on the Shares->Filesystem->Protocols screen to a mode using Kerberos.

The following ports are used by the appliance for Kerberos.

- Kerberos V authentication: 88
- Kerberos V change and set password `SET_CHANGE`: 464
- Kerberos V change and set password `RPCSEC_GSS`: 749

Note: Kerberized NFS clients must access the appliance using an IP address that resolves to an FQDN for those principals. For example, if an appliance is configured with multiple IP addresses, only the IP address that resolves to the appliance's FQDN can be used by its Kerberized NFS clients.

Service Logs

These logs are available for the NFS service:

TABLE 11-7 Logs Available for NFS

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

NFS Analytics

You can monitor NFS activity in the “Analytics” in “Oracle ZFS Storage Appliance Analytics Guide ” section. This includes:

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

Note: When the NFS server reboots or fails over the filename is *unknown* at the server until a new open from the client. The file appears as *unknown* in Analytics worksheets.

NFS BUI and CLI Properties

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

TABLE 11-8 NFS BUI and CLI Properties

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_domain	Custom NFSv4 identity domain
enable_delegation	Enable NFSv4 delegation
mount_visibility	Client share information restriction level
krb5_allow_weak_crypto	Permits weak encryption types (arcfour-hmac-md5-exp, des-cbc-md5, and des-cbc-crc) in Kerberos
krb5_realm	Kerberos Realm
krb5_kdc	Kerberos master KDC
krb5_kdc2	Kerberos slave KDC
krb5_admin	Kerberos admin principal

▼ Sharing a Filesystem over NFS

1. **Go to the Configuration->Services screen.**

2. **Check that the NFS service is enabled and online. If not, enable the service.**

3. **Got to the Chapter 12, "Shares, Projects, and Schema" screen and edit an existing share or create a new share.**

4. **Click the Protocols tab of the share you are editing and check that NFS sharing is enabled. You can also configure the NFS share mode (read/read+write) in this screen.**

iSCSI Service

When you configure a LUN on the appliance you can export that volume over an Internet Small Computer System Interface (iSCSI) target. The iSCSI service allows iSCSI initiators to access targets using the iSCSI protocol.

The service supports discovery, management, and configuration using the iSNS protocol. The iSCSI service supports both unidirectional (target authenticates initiator) and bidirectional (target and initiator authenticate each other) authentication using CHAP. Additionally, the service supports CHAP authentication data management in a RADIUS database.

The system performs authentication first, and authorization second, in two independent steps.

Note - For examples of configuring iSCSI initiators and targets, see the Chapter 6, "Storage Area Network Configuration" section.

iSCSI Service Properties

TABLE 11-9 iSCSI Service 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 "Configuring Services Using the BUI" on page 189and "Configuring Services Using the CLI" on page 192. The CLI property names are shorter versions of those listed above.

iSCSI Service 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.

iSCSI Service Authorization

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

iSCSI Service Targets and Initiators

For more information on iSCSI targets and initiators, see Chapter 6, “Storage Area Network Configuration”.

iSCSI 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.

If, during the failover / failbacks, the iSER Reduced Copy I/Os from the Red Hat client are not surviving:

- Modify the `node.session.timeo.replacement_timeout` parameter in the `/etc/iscsi/iscsid.conf` file to 300sec.

SMB Service

The SMB service provides access to filesystems using the SMB protocol. The supported SMB versions are: SMB1, SMB2.0. Filesystems must be configured to share using SMB from the Chapter 12, “Shares, Projects, and Schema” configuration.

SMB Service Properties

- LAN Manager compatibility level - Authentication modes supported (LM, NTLM, LMv2, NTLMv2). For more information on the supported authentication modes within each compatibility level, consult the Oracle Solaris Information Library for *smb*. NTLMv2 is the recommended minimum security level to avoid publicly known security vulnerabilities.
- Preferred domain controller - The preferred domain controller to use when joining an “Active Directory” on page 242 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.
- Ignore zero VC - When an SMB client establishes a new connection, it may request that the appliance clean up all previous connections and file locks from this client by specifying a Virtual Circuit (VC) number of zero. This protocol artifact however, does not respect network address translation (NAT) for clients or multiple DNS entries assigned to the same host. In combination, zero VC requests between masked or redundant network locations may result in unrelated active connections being reset. By default, zero VC requests are honored to prevent stale file locking, however if SMB sessions are being disconnected in error, ignoring zero VC requests may resolve the issue.

Changing service properties is documented in “Configuring Services Using the BUI” on page 189 and “Configuring Services Using the CLI” on page 192. The CLI property names are shorter versions of those listed above.

SMB Share Properties

Several “Share Properties” on page 281 must be set in certain ways when exporting a share over SMB.

TABLE 11-10 SMB Share Properties

Property	**Description**
Chapter 12, “Shares, Projects, and Schema”	SMB clients expect case-insensitive behavior, so this property must be "mixed"' or "'insensitive".
Chapter 12, “Shares, Projects, and Schema”	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.
“Shares Protocols” on page 311	The name by which clients refer to the share. For information about how this name is inherited from a Chapter 12, “Shares, Projects, and Schema”, see the “Shares Protocols” on page 311 documentation.
“Shares Protocols” on page 311	An ACL which adds another layer of access control beyond the ACLs stored in the filesystem. For more information on this property, see the “Shares Protocols” on page 311 documentation.

The Chapter 12, “Shares, Projects, and Schema” and Chapter 12, “Shares, Projects, and Schema” properties can only be set when creating a share.

NFS/SMB Interoperability

The appliance supports “NFS” on page 195 and SMB clients accessing the same shares concurrently. To correctly configure the appliance for NFS/SMB interoperability, you must configure the following components:

- Configure the “Active Directory” on page 242 service.
- Establish an “Identity Mapping Service” on page 247 strategy and configure the service.
- Configure SMB.
- Configure access control, ACL entries, and ACL inheritance on shares.

SMB and NFSv3 do not use the same access control model. For best results, configure the ACL on the root directory from a SMB client as the SMB access control model is a more verbose model. For information on inheritable trivial ACL entries, see the “Shares > Shares > Access” on page 318 documentation.

SMB DFS Namespaces

The Distributed File System (DFS) is a virtualization technology delivered over the SMB and MSRPC protocols. DFS allows administrators to group shared folders located on different servers by transparently connecting them to one or more DFS namespaces. A DFS namespace is a virtual view of shared folders in an organization. An administrator can select which shared folders to present in the namespace, design the hierarchy in which those folders appear and determine the names that the shared folders show in the namespace. When a user views the namespace, the folders appear to reside in a single, high-capacity file system. Users can navigate the folders in the namespace without needing to know the server names or shared folders hosting the data.

Only one share per system may be provisioned as a standalone DFS namespace. Domain-based DFS namespaces are not supported. Note that one DFS namespace may be provisioned per cluster, even if each cluster node has a separate storage pool. To provision a SMB share as a DFS namespace, use the DFS Management MMC Snap-in to create a standalone namespace.

When the appliance is not joined to an “Active Directory” on page 242 domain, additional configuration is necessary to allow Workgroup users to modify DFS namespaces. To enable an SMB local user to create or delete a DFS namespace, that user must have a separate local account created on the server. In the example below, the steps let the SMB local user `dfsadmin` manipulate DFS namespaces.

SMB Microsoft Stand-alone DFS Namespace Management Tools Support Matrix

The following table lists operations (subcommands/options) of the Microsoft DFS tools on various Windows operating system versions. It identifies which of these are supported by the DFS service on the appliance for managing a standalone DFS namespace on the appliance.

```
Microsoft Windows systems         XP|2003|2003|Vista|2008|2008|Win7|
                                    |    |  R2|     |    |  R2|    |
                                 SP3| SP2| SP2|  SP2| SP2| SP1| SP1|
                                    |    |    |     |    |    |    |
dfscmd CLI:                         |    |    |     |    |    |    |
                                    |    |    |     |    |    |    |
/map [comment] [/restore]          y|   y|   y|    y|   y|   y|   y|
/unmap                             y|   y|   y|    y|   y|   y|   y|
/add [/restore]                    y|   y|   y|    y|   y|   y|   y|
/remove                            y|   y|   y|    y|   y|   y|   y|
/view [/partial | /full]           y|   y|   y|    y|   y|   y|   y|
                                    |    |    |     |    |    |    |
                                    |    |    |     |    |    |    |
dfsutil CLI (old format):           |    |    |     |    |    |    |
                                    |    |    |     |    |    |    |
/addstdroot [/comment]             y|   y|   y|    n|   n|   y|   y|
/remstdroot                        y|   y|   y|    n|   n|   y|   y|
/root:<DfsName> /view              n|   n|   n|    y|   y|   y|   y|
/addlink [/comment]               NA|  NA|  NA|    y|   y|   y|   y|
/removelink                       NA|  NA|  NA|    y|   y|   y|   y|
/state /display                   NA|  NA|  NA|    y|   y|   y|   y|
/state /enable                    NA|  NA|  NA|    y|   y|   y|   y|
/state /disable                   NA|  NA|  NA|    y|   y|   y|   y|
/ttl /display                     NA|  NA|  NA|    y|   y|   y|   y|
/ttl /set                         NA|  NA|  NA|    y|   y|   y|   y|
/server:<MachineName> /view        y|   y|   y|    y|   y|   y|   y|
                                    |    |    |     |    |    |    |
                                    |    |    |     |    |    |    |
dfsutil CLI (new format):           |    |    |     |    |    |    |
                                    |    |    |     |    |    |    |
root addstd [comment]             NA|  NA|  NA|    n|   n|   y|   y|
root remove                       NA|  NA|  NA|    n|   n|   y|   y|
root (view namespace)             NA|  NA|  NA|    y|   y|   y|   y|
link add [comment]                NA|  NA|  NA|    y|   y|   y|   y|
link remove                       NA|  NA|  NA|    y|   y|   y|   y|
link (view)                       NA|  NA|  NA|    y|   y|   y|   y|
target add                        NA|  NA|  NA|    y|   y|   y|   y|
target remove                     NA|  NA|  NA|    y|   y|   y|   y|
target (view)                     NA|  NA|  NA|    y|   y|   y|   y|
property comment (view)           NA|  NA|  NA|    y|   y|   y|   y|
property comment set              NA|  NA|  NA|    y|   y|   y|   y|
property ttl (view)               NA|  NA|  NA|    y|   y|   y|   y|
property ttl set                  NA|  NA|  NA|    y|   y|   y|   y|
property state (view)             NA|  NA|  NA|    y|   y|   y|   y|
property state offline            NA|  NA|  NA|    y|   y|   y|   y|
property state online             NA|  NA|  NA|    y|   y|   y|   y|
                                    |    |    |     |    |    |    |
                                    |    |    |     |    |    |    |
DFS GUI:                            |    |    |     |    |    |    |
                                    |    |    |     |    |    |    |
add standalone root                y|   y|   y|    n|   n|   n|   n|
remove standalone root             y|   y|   y|    n|   n|   n|   n|
change root comment                y|   y|   y|    n|   n|   n|   n|
change root timeout                y|   y|   y|    n|   n|   n|   n|
add link                           y|   y|   y|    n|   n|   n|   n|
remove link                        y|   y|   y|    n|   n|   n|   n|
change link comment                y|   y|   y|    n|   n|   n|   n|
change link timeout                y|   y|   y|    n|   n|   n|   n|
add link's target                  y|   y|   y|    n|   n|   n|   n|
remove link's target               y|   y|   y|    n|   n|   n|   n|
enable link's referral (target)    y|   y|   y|    n|   n|   n|   n|
disable link's referral (target)   y|   y|   y|    n|   n|   n|   n|
hide root                          y|   y|   y|    y|   y|   y|   y|
show root                          y|   y|   y|    y|   y|   y|   y|
display links                      y|   y|   y|    n|   n|   n|   n|
display targets                    y|   y|   y|    n|   n|   n|   n|
                                  XP|2003|2003|Vista|2008|2008|Win7|
                                    |    |  R2|     |    |  R2|    |
                                 SP3| SP2| SP2|  SP2| SP2| SP1| SP1|
```

Notes: y - supported n - not supported NA - not applicable

- Solaris does not verify the DFS link target.
- CLI commands for modifying and viewing comment and timeout (TTL) are applicable to both root and link.
- CLI commands for viewing state are applicable to root, root's target, link, and link's target.
- CLI commands for modifying state are only applicable for link and link's target.

▼ Example: Manipulating 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.**

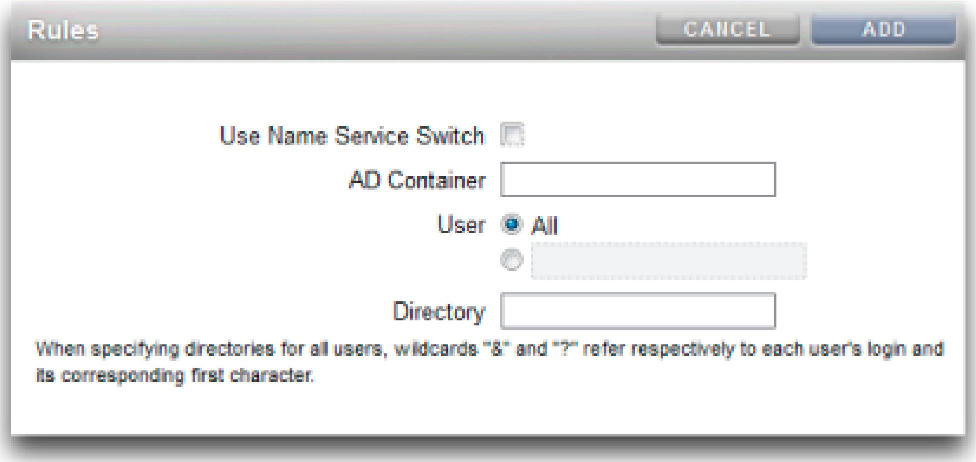
2. **Add `dfsadmin` to the local SMB group Administrators.**

3. **Login as `dfsadmin` on the Windows machine from which the DFS namespace will be modified.**

SMB Autohome Service

For Windows file sharing, Autohome provides access to filesystems using the SMB protocol. Autohome defines and maintains home directory shares for users that access the system through SMB. Autohome rules map SMB clients to home directories.

FIGURE 11-2 Setting Autohome Rules



Use Name Service Switch □

AD Container

User ● All
○

Directory

When specifying directories for all users, wildcards "&" and "?" refer respectively to each user's login and its corresponding first character.

- Use Name Service Switch - Toggles Name Service Switch (NSS) on or off. You cannot create an NSS rule and an rule for all users at the same time.
- AD Container - Sets the Active Directory container, for example: dc=com,dc=fishworks, ou=Engineering,CN=myhome.
- User - Sets the Autohome rule for all All users or for the user you specify. When you specify a user, the wildcards "&" and "?" refer to a user's login and its corresponding first character.
- Directory - Sets the directory for the rule, for example: /export/wdp.

▼ Adding SMB Autohome Rules

1. **Use the `create` command to add autohome rules, and the `list` command to list existing rules. This example adds a rule for the user "Bill" then lists the rules:**

```
twofish:> configuration services smb
twofish:configuration services smb> create
twofish:configuration services rule (uncommitted)> set use_nss=false
twofish:configuration services rule (uncommitted)> set user=Bill
twofish:configuration services rule (uncommitted)> set directory=/export/wdp
twofish:configuration services rule (uncommitted)> set container="dc=com,dc=fishworks,
   ou=Engineering,CN=myhome"
twofish:configuration services rule (uncommitted)> commit
twofish:configuration services smb> list
RULE       NSS      USER         DIRECTORY            CONTAINER
rule-000   false    Bill         /export/wdp          dc=com,dc=fishworks,
   ou=Engineering,CN=myhome
```

2. **Autohome rules may be created using wildcard characters. The *&* character matches the users' username, and the *?* character matches the first letter of the users' username. The following uses wildcards to match all users:**

```
twofish:configuration services smb> create
twofish:configuration services rule (uncommitted)> set use_nss=false
twofish:configuration services rule (uncommitted)> set user=*
twofish:configuration services rule (uncommitted)> set directory=/export/?/&
twofish:configuration services rule (uncommitted)> set container="dc=com,dc=fishworks,
   ou=Engineering,CN=myhome"
twofish:configuration services rule (uncommitted)> commit
twofish:configuration services smb> list
RULE       NSS      USER         DIRECTORY            CONTAINER
rule-000   false    Bill         /export/wdp          dc=com,dc=fishworks,
   ou=Engineering,CN=myhome
```

3. **The name service switch may also be used to create autohome rules:**

```
twofish:configuration services smb> create
twofish:configuration services rule (uncommitted)> set use_nss=true
twofish:configuration services rule (uncommitted)> set container="dc=com,dc=fishworks,
   ou=Engineering,CN=myhome"
twofish:configuration services rule (uncommitted)> commit
twofish:configuration services smb> list
RULE       NSS      USER         DIRECTORY            CONTAINER
rule-000   true                                       dc=com,dc=fishworks,
   ou=Engineering,CN=myhome
```

SMB Local Groups

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

TABLE 11-11 SMB Local Groups

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.

▼ Adding a User to an SMB Local Group

● To add a user, do the following:

```
twofish:configuration services smb> groups
twofish:configuration services smb groups> create
twofish:configuration services smb member (uncommitted)> set user=Bill
twofish:configuration services smb member (uncommitted)> set group="Backup Operators"
twofish:configuration services smb member (uncommitted)> commit
twofish:configuration services smb groups> list
MEMBER       USER                         GROUP
member-000   WINDOMAIN\Bill               Backup Operators
```

SMB Local Accounts

Local accounts and user IDs are mapped to Windows user IDs. Note that the *guest* account is a special, readonly account and cannot be configured for read/write in the appliance.

SMB MMC Integration

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

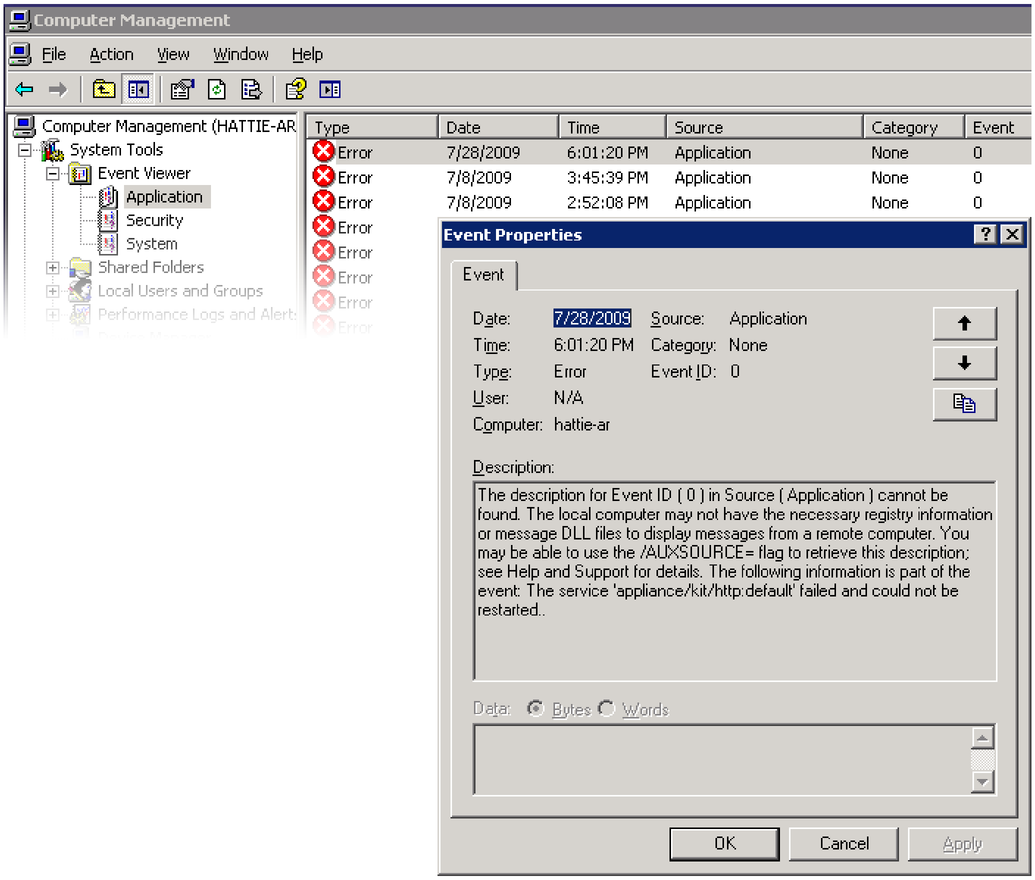
In order to use the MMC functionality on the Sun ZFS Storage 7000 appliances in workgroup mode, be sure to add the Windows administrator who will use the management console to the Administrators “local group” on page 202 on the appliance. Otherwise you may receive an `Access is denied` or similar error 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:

SMB 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.

FIGURE 11-3 SMB Event Viewer



Type	Date	Time	Source	Category	Event
Error	7/28/2009	6:01:20 PM	Application	None	0
Error	7/8/2009	3:45:39 PM	Application	None	0
Error	7/8/2009	2:52:08 PM	Application	None	0
Error					
Error					
Error					
Error					

Date: 7/28/2009 Source: Application
Time: 6:01:20 PM Category: None
Type: Error Event ID: 0
User: N/A
Computer: hattie-ar

Description:

The description for Event ID (0) in Source (Application) cannot be found. The local computer may not have the necessary registry information or message DLL files to display messages from a remote computer. You may be able to use the /AUXSOURCE= flag to retrieve this description; see Help and Support for details. The following information is part of the event: The service 'appliance/kit/http:default' failed and could not be restarted..

Data: ● Bytes ○ Words

SMB Share Management

Support for share management includes the following:

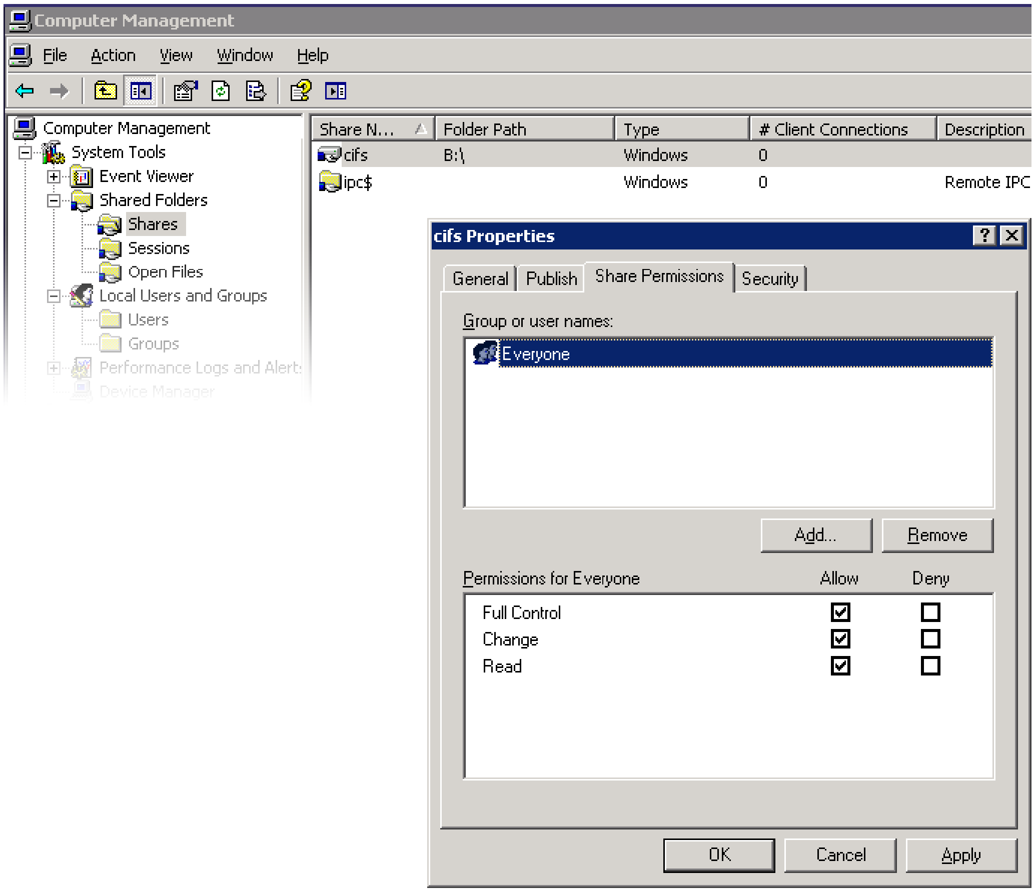
- Listing shares

- Setting ACLs on shares
- Changing share permissions
- Setting the description of a share

Features not currently supported via MMC include the following:

- Adding or Deleting a share
- Setting client side caching property
- Setting maximum allowed or number of users property

FIGURE 11-4 SMB Share Permission Properties



Share N...	Folder Path	Type	# Client Connections	Description
cifs	B:\	Windows	0	
ipc$		Windows	0	Remote IPC

Group or user names:

Everyone

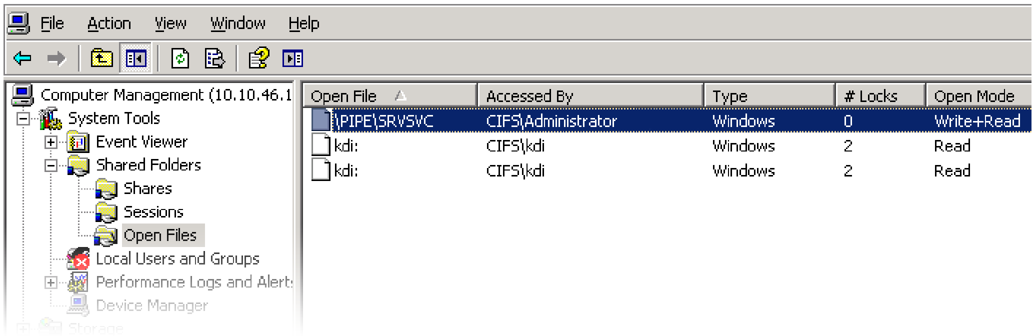
Permissions for Everyone	Allow	Deny
Full Control	☑	□
Change	☑	□
Read	☑	□

SMB Users, Groups, and Connections

The following features are supported:

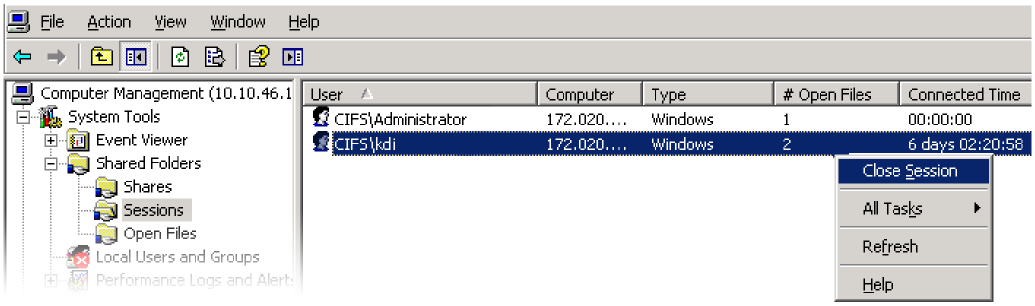
- Viewing local SMB users and groups
- Listing user connections, including listing the number of open files per connection
- Closing user connections
- Listing open files, including listing the number of locks on the file and file open mode
- Closing open files

FIGURE 11-5 Open Files per Connection



Open File	Accessed By	Type	# Locks	Open Mode
\PIPE\SRVSVC	CIFS\Administrator	Windows	0	Write+Read
kdi:	CIFS\kdi	Windows	2	Read
kdi:	CIFS\kdi	Windows	2	Read

FIGURE 11-6 Open Sessions

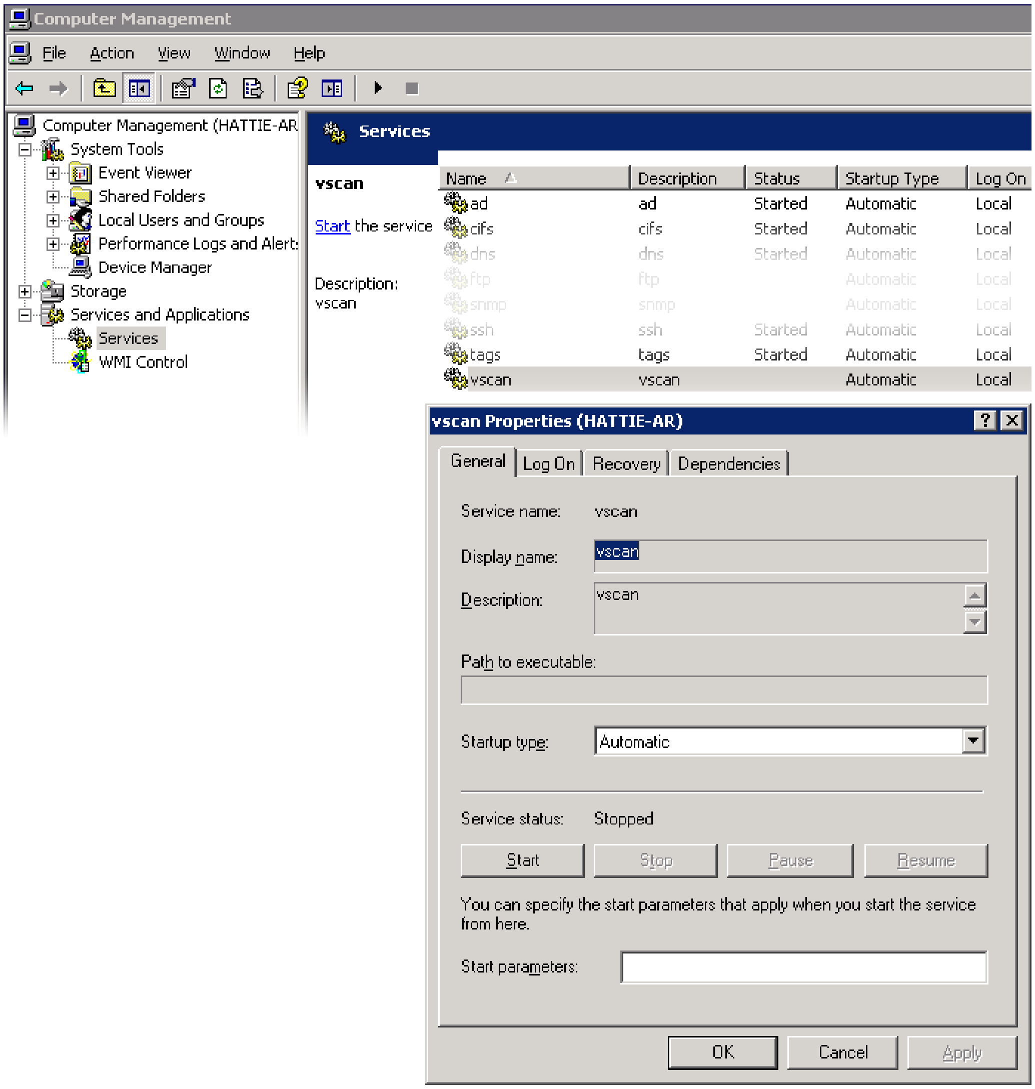


User	Computer	Type	# Open Files	Connected Time
CIFS\Administrator	172.020....	Windows	1	00:00:00
CIFS\kdi	172.020....	Windows	2	6 days 02:20:58

Listing SMB Services

Support includes listing of ZFSSA services. 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.

FIGURE 11-7 vscan Properties



vscan

Description:
vscan

Name	Description	Status	Startup Type	Log On
ad	ad	Started	Automatic	Local
cifs	cifs	Started	Automatic	Local
dns	dns	Started	Automatic	Local
ftp	ftp		Automatic	Local
snmp	snmp		Automatic	Local
ssh	ssh	Started	Automatic	Local
tags	tags	Started	Automatic	Local
vscan	vscan		Automatic	Local

Service name: vscan

Display name: vscan

Description: vscan

Path to executable:

Startup type: Automatic

Service status: Stopped

You can specify the start parameters that apply when you start the service from here.

Start parameters:

To ensure that only the appropriate users have access to administrative operations there are some access restrictions on the operations performed remotely using MMC.

TABLE 11-12 Users and Allowed Operations

USERS	ALLOWED OPERATIONS
Regular users	List shares.
Members of the Administrators or Power Users groups	Manage shares, list user connections.
Members of the Administrators group	List open files and close files, disconnect user connections, view services and event log.

Configuring SMB Using the BUI

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 Oracle.**

25. **Enter your Online Account user name and password. A privacy statement will be displayed for your review.**

26. **Choose which of your inventory teams to register with. The default team for each account is the same as the account user name, prefixed with a '$'.**

27. **Commit your initial configuration changes.**

▼ 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 Service

The FTP (File Transfer Protocol) service allows filesystem access from FTP clients. Anonymous logins are not allowed, users must authenticate with whichever name service is configured in Services.

FTP Properties

FTP General Settings

TABLE 11-13 FTP 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

Property	Description
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
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

FTP Security Settings

TABLE 11-14 FTP 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
Permit foreign data connection addresses	Permits foreign FTP connections to enable direct transfer of files between FTP servers. This property is off by default.

Changing services properties is documented in “Configuring Services Using the BUI” on page 189 and “Configuring Services Using the CLI” on page 192. The CLI property names are shorter versions of those listed above.

FTP Logs

TABLE 11-15 FTP 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

Configuring FTP Using the BUI

▼ Allowing FTP Access to a share

1. **Go to Configuration->Services**
2. **Ensure that the FTP service is enabled and online. If not, enable the service.**
3. **Select or add a share in the Shares screen.**
4. **Go to the "Protocols" section, and check that FTP access is enabled. This is also where the mode of access (read/read+write) can be set.**

HTTP Service

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.

HTTP Properties

TABLE 11-16 HTTP 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.
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 “Configuring Services Using the BUI” on page 189 and “Configuring Services Using the CLI” on page 192. The CLI property names are shorter versions of those listed above.

HTTP Authentication and Access Control

If the "Require client login" option is enabled, then the appliance will deny access to clients that do not supply valid authentication credentials for a local user, a NIS user, or an LDAP user. Active Directory authentication is not supported.

Only basic HTTP authentication is supported. Note that unless HTTPS is being used, this transmits the username and password unencrypted, which may not be appropriate for all environments.

Normally, authenticated users have the same permissions with HTTP that they would have with NFS or FTP. Files and directories created by an authenticated user will be owned by that user, as viewed by other protocols. Privileged users (those having a uid less than 100) will be treated as "nobody" for the purposes of access control. Files created by privileged users will be owned by "nobody".

If the "Require client login" option is disabled, then the appliance will not try to authenticate clients (even if they do supply credentials). Newly created files are owned by "nobody", and all users are treated as "nobody" for the purposes of access control.

Regardless of authentication, no permissions are masked from created files and directories. Created files have Unix permissions 666 (readable and writable by everyone), and created directories have Unix permissions 777 (readable, writable, and executable by everyone).

HTTP Logs

TABLE 11-17 HTTP Logs

Log	Description
network-http:apache22	HTTP service log

Configuring HTTP

▼ 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 Service

The NDMP (Network Data Management Protocol) service enables the system to participate in NDMP-based backup and restore operations controlled by a remote NDMP client called a Data Management Application (DMA). Using NDMP, appliance user data (i.e., data stored in administrator-created shares on the appliance) can be backed up and restored to both locally attached tape devices and remote systems. Locally-attached tape devices can also be exposed to the DMA for backing up and restoring remote systems.

NDMP cannot be used to backup and restore system configuration data. Instead, use the [[Maintenance:System:ConfigurationBackup|Configuration Backup and Restore]] feature.

NDMP Local vs. Remote Configurations

The appliance supports backup and restore using both a *local* configuration, in which tape drives are physically attached to the appliance, and a *remote* configuration, in which data is streamed to another system on the same network. In both cases, the backup must be managed by a supported DMA.

In local configurations, supported tape devices, including both drives and changers (robots), are physically connected to the system using a supported SCSI or Fibre Channel (FC) card configured in Initiator mode. These devices can be viewed on the “NDMP status” on page 60 screen. The NDMP service presents these devices to a DMA when the DMA scans for devices. Once configured in the DMA, these devices are available for backup and restore of the appliance or other systems on the same network. After adding tape drives or changers to the system or removing such devices from the system, a reboot may be required before the changes will be recognized by the NDMP service. After that, the DMA may need to be reconfigured because tape device names may have changed.

In remote configurations, the tape devices are not physically connected to the system being backed up and restored (the data server) but rather to the system running the DMA or a separate system (the tape server). These are commonly called "3-way configurations" because the DMA controls two other systems. In these configurations the data stream is transmitted between the data server and the tape server over an IP network.

NDMP Backup Formats and Types

The NDMP protocol does not specify a backup data format. The appliance supports three backup types corresponding to different implementations and on-tape formats. DMAs can select a backup type using the following values for the NDMP environment variable "TYPE":

TABLE 11-18 NDMP Backup Formats and Types

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.

NDMP Back up with "dump" and "tar"

When backing up with "dump" and "tar" backup types, administrators specify the data to backup by a filesystem path, called the *backup path*. For example, if the administrator configures a backup of */export/home*, then the share mounted at that path will be backed up. Similarly, if a backup stream is restored to */export/code*, then that's the path where files will be restored, even if they were backed up from another path.

Only paths which are mountpoints of existing shares or contained within existing shares may be specified for backup. If the backup path matches a share's mountpoint, only that share is backed up. Otherwise the path must be contained within a share, in which case only the portion of that share under that path is backed up. In both cases, other shares mounted inside the specified share under the backup path will not be backed up; these shares must be specified separately for backup.

Snapshots - If the backup path specifies a live filesystem (e.g., */export/code*) or a path contained within a live filesystem (e.g., */export/code/src*), the appliance immediately takes a new snapshot and backs up the given path from that snapshot. When the backup completes, the snapshot is destroyed. If the backup path specifies a snapshot (e.g., */export/code/.zfs/snapshot/mysnap*), no new snapshot is created and the system backs up from the specified snapshot.

Share metadata - To simplify backup and restore of complex share configurations, "dump" and "tar" backups include share metadata for projects and shares associated with the backup path. This metadata describes the share configuration on the appliance, including protocol sharing properties, quota properties, and other properties configured on the Shares screen. This is not to be confused with filesystem metadata like directory structure and file permissions, which is also backed up and restored with NDMP.

For example, if you back up /export/proj, the share metadata for all shares whose mountpoints start with /export/proj will be backed up, as well as the share metadata for their parent projects. Similarly, if you back up /export/someshare/somedir, and a share is mounted at /export/someshare, that share and its project's share metadata will be backed up.

When restoring, if the destination of the restore path is not contained inside an existing share, projects and shares in the backup stream will be recreated as needed with their original properties as stored in the backup. For example, if you back up /export/foo, which contains project proj1 and shares share1 and share2, and then destroy the project and restore from the backup, then these two shares and the project will be recreated with their backed-up properties as part of the restore operation.

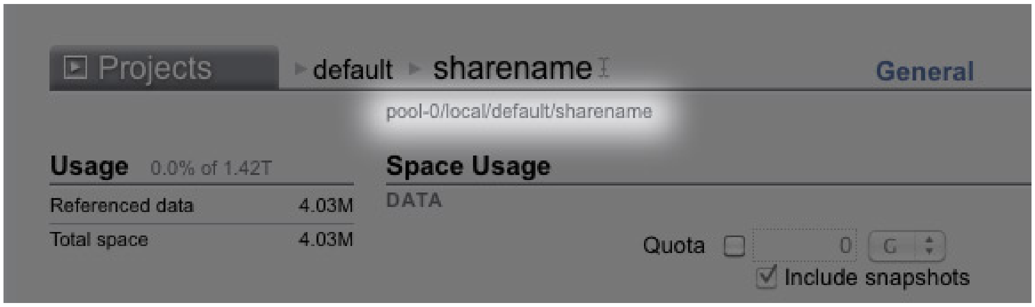
During a restore, if a project exists that would have been automatically recreated, the existing project is used and no new project is automatically created. If a share exists that would have been automatically recreated, and if its mountpoint matches what the appliance expects based on the original backup path and the destination of the restore, then the existing share is used and no new share is automatically created. Otherwise, a new share is automatically created from the metadata in the backup. If a share with the same name already exists (but has a different mountpoint), then the newly created share will be given a unique name starting with "ndmp-" and with the correct mountpoint.

It is recommended that you either restore a stream whose datasets no longer exist on the appliance, allowing the appliance to recreate datasets as specified in the backup stream, or precreate a destination share for restores. Either of these practices avoids surprising results related to the automatic share creation described above.

NDMP Back up with "zfs"

When backing up with type "zfs", administrators specify the data to backup by its canonical name on the appliance. This can be found underneath the name of the share in the BUI:

FIGURE 11-8 NDMP Share Name



or in the CLI as the value of the canonical_name property. Canonical names do not begin with a leading '/', but when configuring the backup path the canonical name must be prefixed with '/'.

Both projects and shares can be specified for backup using type "zfs". If the canonical name is specified as-is, then a new snapshot is created and used for the backup. A specific snapshot can be specified for backup using the '@snapshot' suffix, in which case no new snapshot is created and the specified snapshot is backed up. For example:

TABLE 11-19 Canonical Names and Shares Backed Up

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 filesystem or volume.

Canonical name	Shares backed up
pool-0/local/default/code@yesterday	Named snapshot "yesterday" of share "code" in local project "default". "code" could be 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.

NDMP Incremental backups

The appliance supports level-based incremental backups for all of the above backup types. To specify a level backup, DMAs typically specify the following three environment variables:

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:

TABLE 11-20 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).

NDMP Properties

The NDMP service configuration consists of the following properties:

TABLE 11-21 NDMP Properties

Property	Description
Version	The version of NDMP that your DMA supports.
TCP port (v4 only)	The NDMP default connection port is 10000. NDMPv3 always uses this port. NDMPv4 allows a different port if needed.
Default restore pool(s)	When you perform a full restore using "tar" or "dump", the system re-creates datasets if there is no share mounted at the target. Because the NDMP protocol specifies only the mount point, the system chooses a pool in which to recreate projects and shares. On a system with multiple pools, this property lets you specify one or more pools. Multiple pools only need to be specified in a cluster with active pools on each head. You must ensure that this list is kept in sync with any storage configuration changes. If none of the pools exist or are online, the system will select a default pool at random.
Ignore metadata-only changes	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.
Allow token-based backup	Enables or disables token-based method for ZFS backup. This property is off by default.
ZFS rollback before restore (v4 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 data sets 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.
Allow direct access recovery	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

Property	Description
	this option at backup time in order to be able to recover individual files later.
Restore absolute paths (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.
DMA tape mode (for locally attached drives)	Specifies whether the DMA expects System V or BSD semantics. The default is System V, 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.
DMA username and password	Used to authenticate the DMA. The system uses MD5 for user authentication

Changing services properties is documented in “Configuring Services Using the BUI” on page 189 and “Configuring Services Using the CLI” on page 192. The CLI property names are shorter versions of those listed above.

NDMP Logs

TABLE 11-22 NDMP Logs

Log	Description
system-ndmpd:default	NDMP service log

Remote Replication

The remote replication service facilitates replication of projects and shares to and from other Oracle ZFS Storage Appliances. This functionality is described in detail in the Chapter 13, “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 Chapter 13, "Replication" documentation.

Shadow Migration

The shadow migration service allows for automatic migration of data from external or internal sources. This functionality is described in great detail in Chapter 14, "Shadow Migration". 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.

Shadow Migration Properties

TABLE 11-23 Shadow 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 "Configuring Services Using the BUI" on page 189 and "Configuring Services Using the CLI" on page 192. The CLI property names are shorter versions of those listed above.

SFTP Service

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.

SFTP Properties

- Port (for incoming connections) - The port SFTP listens on. Default is 218

- Permit root login - Allows SFTP logins for the root user. This is off by default.
- Logging level - The verbosity of SFTP log messages
- SFTP Keys - RSA/DSA public keys for SFTP authentication. Text comments can be associated with the keys to help administrators track why they were added. As of the 2011.1 software release, key management for SFTP has changed to increase security. When creating an SFTP key, it is required to include the "user" property with a valid user assignment. SFTP keys are grouped by user and are authenticated via SFTP with the user's name. It is recommended to recreate any existing SFTP keys that do not include the user property, even though they will still authenticate.

Changing services properties is documented in "Configuring Services Using the BUI" on page 189 and "Configuring Services Using the CLI" on page 192. 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
```

SFTP Logs

TABLE 11-24 SFTP Logs

Log	Description
network-sftp:default	Logs SFTP service events

Configuring SFTP

▼ 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.**

▼ Configuring SFTP Services for Remote Access

1. **Create a local user or network user (LDAP or NIS) with an appropriate administrator role. (See Chapter 7, "User Configuration").**

2. **Generate an SSH authentication key by entering the command `ssh-keygen -t dsa` on the Solaris host/client.**

3. **Enter a file name in which to store the key.**

4. **Enter a passphase if required, or leave this field blank to log on directly to the SFTP share. The location is displayed for the key. The key looks similar to the following:**

5. **: ssh-dss AAAAB3NzaC1kc3MAAACBAPMMs5h8UWk1NPf/VJDDEo0OAwT+s6iZxkCmmrgAmLfTX9izWk+**

6. **: bsvNldOlXN/6EgkusLjo/+UaEt5+704vMHClRaq3AlVHLS5tVjeX3iCs+fDo0qwXZg3Brh8QBAaWk3**

7. **:ywr2osulI1tHh4v/HwEAHZq5mVWXav0pO3bgmxl0/+VAAAAFQDIJxnm52DfyEdQQMTY+jRVvzGwMQA**

8. **: AAIAhTP6Ey+2gGFiCKkvUofsco4d8pbqH8duE9P6Y88s0+opuj52GkAdRUt2fRrdM9Cf3h4lIOc8Bw9**

9. **: bZIBzrCKBNWBUdZG56tsfLdilW6vS6gxKrmL2v7fSp9WYPsxZGhOLfU29zW4n2WVcVHbGyFEoVe+taq**

10. **: aq+AYJaWoHnjZL1/LpQAAAIAOLc8+uc3hDOcK3pAkYdg8b2rYIGOAZU4py0rq24DGPeVHd5h5jbe4p**

11. **:WDM70uYqGCOPYiOKeEoMNJpczRX5qjI+BfoUY4sH24WWwsKkT8XX9PUAa0WT+7axEqg2N6YelaTJ95J**

12. **:vMaj6E7HkAIra2Sj2H/LSDktL42UL+j1Wx5A== username sunray**

13. **Go to Configuration > Services > SFTP. Under Keys, click the plus (+) sign.**

14. **In the New Key window, select DSA.**

15. **Copy only the key portion (beginning with AAAA and ending with Wx5A== in the example above) and paste into the Key field. Enter the user name and add a comment as a reminder.**

16. **: Note: The key should not contain any white spaces.**

17. **Go to Shares > Shares and click the plus (+) sign to create a filesystem.**

18. **In the Create Filesystem window, enter the filesystem name (for example, sftp), change the permissions to Read/ Write for the share, and click Apply.**

19. **Click the pencil icon to set up the share properties. (See Chapter 12, “Shares, Projects, and Schema”.)**

20. **To access the share, use the `sftp` command as shown in these examples:**

21. **: sftp -o "port=218" <username> 10.x.x.151:/export/sftp**

22. **: Connecting to 10.x.xx.151...**

23. **: Changing to: /export/sftp**

24. **: sftp>**

25. **: Example with -v option:**

26. **: sftp -v -o "IdentityFile=/home/<username>/.ssh/id_dsa" -o "port=218"**

27. **: root 10.x.xx.151:/export/sftp**

SRP Service

When you configure a LUN on the appliance you can export that volume over a SCSI Remote Protocol (SRP) target. The SRP service allows initiators to access targets using the SRP protocol.

For information on SRP targets and initiators, see Chapter 6, “Storage Area Network Configuration”.

For examples of administering SRP targets, see Chapter 6, “Storage Area Network Configuration”.

TFTP Service

Trivial File Transfer Protocol (TFTP) is a simple protocol to transfer files. TFTP is designed to be small and easy to implement, therefore, lacks most of the features of a regular FTP. TFTP only reads and writes files (or mail) from/to a remote server. It cannot list directories, and currently has no provisions for user authentication..

TFTP Properties

TABLE 11-25 TFTP Properties

Property	Description
Default Root Directory	The TFTP login location. The default is "/export" and points to the top of the shares hierarchy. All users will be logged into this location after successfully authenticating with the TFTP service

Changing services properties is documented in "Configuring Services Using the BUI" on page 189 and "Configuring Services Using the CLI" on page 192. The CLI property names are shorter versions of those listed above.

Configuring TFTP

▼ Allowing TFTP access to a share

1. **Go to Configuration->Services**
2. **Check that the TFTP service is enabled and online. If not, enable the service.**
3. **Select or add a share in the Shares screen.**
4. **Go to the "Protocols" section, and check that TFTP access is enabled. This is also where the mode of access (read/read+write) can be set.**

Virus Scan Service

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.

Virus Scan Properties

TABLE 11-26 Virus Scan 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 “Configuring Services Using the BUI” on page 189 and “Configuring Services Using the CLI” on page 192. The CLI property names are shorter versions of those listed above.

Virus Scan File Extensions

This section describes how to control which files are scanned. The default value, " * ", causes all files to be scanned. Scanning all files may impact performance so you can designate a subset of files to scan.

For example, to scan only high-risk files, including zip files, but not files with names that match the pattern "data-archive*.zip", you could configure the following settings:

TABLE 11-27 Virus Scan File Extensions

Action	Pattern
Scan	exe
Scan	com
Scan	bat

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

Note: You must use "Don't Scan *" to exclude all other file types not explicitly included in the scan list. A file named "file.name.exe.bat.jpg123" would NOT be scanned, as only the "jpg123" portion of the name, the extension, would be compared against the rules.

Do NOT use exclude settings before include settings. For example, do not use a "Don't Scan *" setting before include settings since that would exclude all file types that come after it. The following example would not scan any files:

TABLE 11-28 Virus Scan Actions

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

Scanning Engines

In this section, specify which scanning engines to use. A scanning engine is an external third-party virus scanning server which the appliance contacts using ICAP (Internet Content Adaptation Protocol, RFC 3507) to have files scanned.

TABLE 11-29 Scanning Engines Properties

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.

Property	Description
Port	Port for the scan engine

Virus Scan Logs

TABLE 11-30 Virus Scan Logs

Log	Description
vscan	Log of the Virus Scan service

Configuring Virus Scan

▼ 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 Service

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 Service" on page 217 and "HTTP Service" on page 219.
- NIS users can be granted privileges for appliance administration. The appliance supplements NIS information with its own privilege settings.

Note that UIDs and GIDs from 0-99 inclusive are reserved by the operating system vendor for use in future applications. Their use by end system users or vendors of layered products is not supported and may cause security related issues with future applications.

NIS Properties

TABLE 11-31 NIS 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 "Configuring Services Using the BUI" on page 189 and "Configuring Services Using the CLI" on page 192. The CLI property names are shorter versions of those listed above.

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

NIS Logs

TABLE 11-32 NIS 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

Configuring NIS

▼ 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 Chapter 7, “User Configuration” for adding authorizations to this user.**

LDAP Service

LDAP (Lightweight Directory Access Protocol) is a directory service for centralizing management of users, groups, hostnames and other resources (called objects). This service on the appliance acts as an LDAP client so that:

- LDAP users can log in to “FTP Service” on page 217 and “HTTP Service” on page 219.
- LDAP user names (instead of numerical ids) can be used to configure root directory ACLs on a share.
- LDAP users can be granted privileges for appliance administration. The appliance supplements LDAP information with its own privilege settings.
- The LDAP server's certificate can be self-signed.
- You cannot supply a list of trusted CA certificates; each certificate must be individually accepted by the appliance administrator.
- When an LDAP server's certificate expires, you must delete the server from the list and then re-add it to accept its new certificate.

Note UIDs from 0-99 inclusive are reserved by the operating system vendor for use in future applications. Their use by end system users or vendors of layered products is not supported and can cause security issues with other applications.

LDAP Properties

For the appropriate settings for your environment, consult your LDAP server administrator.

- Protect LDAP traffic with SSL/TLS - Toggles TLS (Transport Layer Security, the descendant of SSL) to establish secure connections to the LDAP server

- Base search DN - Supplies the distinguished name of the base object which is the starting point for directory searches.
- Search scope - Defines which objects in the LDAP directory are searched, relative to the base object. Search results can be limited only to objects directly beneath the base search object (one-level) or they can include any object beneath the base search object (subtree). The default is one-level.
- Authentication method - Method used to authenticate the appliance to the LDAP server. The appliance supports Simple (RFC 4513), SASL/DIGEST-MD5, and SASL/GSSAPI authentication. If the Simple authentication method is used, SSL/TLS should be enabled so the user's DN and password are not sent in plain text. When using the SASL/GSSAPI authentication method, only the self bind credential level is available.
- Bind credential level - Credentials used to authenticate the appliance to the LDAP server.
- * Anonymous gives the appliance access only to data that is available to everyone.
- * Proxy directs the service to bind via a specified account.
- * Proxy DN - Distinguished name of account used for proxy authentication.
- * Proxy Password - Password for account used for proxy authentication.
- * Self - Self authenticates the appliance using the user's identity and credentials. Self authentication can only be used with the SASL/GSSAPI authentication method.
- Schema definition - Schema used by the appliance. This property lets administrators override the default search descriptor, attribute mappings, and object class mappings for users, groups, and netgroups. For more information, see "LDAP Service" on page 238.
- Servers - List of LDAP servers to use. If only one server is specified, the appliance uses only that server and LDAP services are unavailable if that server fails. If multiple servers are specified, any functioning server can be used at any time without preference. If any server fails, another server in the list is used. LDAP services remain available unless all specified servers fail.

LDAP Custom Mappings

To look up users and groups in the LDAP directory, the appliance uses a search descriptor and must know which object classes correspond to users and groups and which attributes correspond to the properties needed. By default, the appliance uses object classes specified by RFC 2307 (*posixAccount* and *posixGroup*) and the default search descriptors shown in the following list, but this can be customized for different environments. The base search DN used in the examples below is *dc=example,dc=com*:

TABLE 11-33 LDAP Custom Mappings

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

Search descriptor	Default value	Example
netgroups	ou=netgroup,*base search DN*	ou=netgroup,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, group, and netgroup attributes and objects, choose the appropriate tab ("Users", "Groups", or "Netgroups") and specify mappings using the *default* = *new* syntax, where *default* is the default value and *new* is the value you want to use. For examples:

- To use *unixaccount* instead of *posixAccount* as the user object class, enter posixAccount = unixaccount in Object class mappings on the Users tab.
- To use *employeenumber* instead of *uid* as the attribute for user objects, enter uid = employeenumber in Attribute mappings on the Users tab.
- To use *unixgroup* instead of *posixGroup* as the group object class, type posixGroup = unixgroup in Object class mappings on the Groups tab.
- To use *groupaccount* instead of *cn* as the attribute for group objects, enter cn = groupaccount in Attribute mappings on the Groups tab.

The following is a list of object classes and attributes that you might want to map:

- Classes:
- * posixAccount
- * posixGroup
- * shadowAccount
- Attributes - Users:
- * uid
- * uidNumber
- * gidNumber
- * gecos
- * homeDirectory
- * loginShell
- * userPassword
- Attributes - Groups:
- * uid
- * memberUid
- * cn
- * userPassword
- * gidNumber
- * member
- * uniqueMember

- * memberOf
- * isMemberOf

LDAP Logs

The following is an example log.

TABLE 11-34 LDAP Logs

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

Configuring LDAP

▼ Adding an appliance administrator

To let an existing LDAP user log in using LDAP credentials and administer the appliance, use the following procedure:

1. **On the Configuration => Services => LDAP page, enter the properties that you want to use. For information about the available properties, see "LDAP Properties" on page 238.**

2. **To apply properties you selected, click Apply or click Revert to start over.**

3. **To add LDAP servers, in the Servers section click the add ⊕ icon.** **For information about servers, see the Servers section in "LDAP Properties" on page 238.**

4. **To configure the LDAP server, in the New LDAP Server box, enter the LDAP server Address and select the LDAP Certificate source that you want to use. For the Certificate source, selecting Server searches the current server and retrieves the certificate (in an insecure manner) and uses it in the future to validate the certificate presented later.**

5. **On the Configuration => Users page, add users as needed using LDAP usernames. For information about adding users, see Chapter 7, "User Configuration".**

Active Directory

The Active Directory service provides access to a Microsoft Active Directory database, which stores information about users, groups, shares, and other shared objects. This service has two modes: domain and workgroup mode, which dictate how "SMB" on page 202 users are authenticated. When operating in domain mode, "SMB" on page 202 clients are authenticated through the AD domain controller. In workgroup mode, "SMB" on page 202 clients are authenticated locally as local users. See "Users" for more information on local users.

Active Directory Properties

Active Directory Join Domain

If an account does not already exist in Active Directory by default, a machine trust account for the system is automatically created in the default container for computer accounts (cn=Computers) as part of the domain join operation. The following users are allowed to perform domain join:

- Domain administrator. Can join any number of systems to the domain with machine trust accounts placed in any containers.
- Delegated administrator with authority over one or more Organizational Units. Can join any number of systems to a domain with machine account location designated in the Organizational Units they are responsible for.
- Normal user with machine accounts pre-staged by administrator. Can join a system to the domain as pre-authorized by an administrator.
- Normal user. Normally authorized to join a limited number of systems.

The following properties for joining an Active Directory domain are available:

- Active Directory Domain - The fully-qualified name or NetBIOS name of an Active Directory domain
- User - An AD user who has credentials to create a computer account in Active Directory
- Password - The administrative user's password
- Additional DNS Search Path - When this optional property is specified, DNS queries are resolved against this domain, in addition to the primary DNS domain and the Active Directory domain
- Organizational Unit - Specifies an alternative organizational unit in which the system's machine trust account will be created. The organizational unit is specified as a comma-separated list of one or more name-value pairs using the domain-relative distinguished name (DN) format, for example, ou=innerOU,ou=outerOU.
- Use Pre-created Account - If the system's account exists and the specified Organizational Unit is not the one that the account is in, use the pre-created account.

Active Directory Join Workgroup

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

- Windows Workgroup - A workgroup

Changing services properties is documented in “Configuring Services Using the BUI” on page 189 and “Configuring Services Using the CLI” on page 192. The CLI property names are shorter versions of those listed above.

Active Directory Domains and Workgroups

Instead of enabling and disabling the service directly, the service is modified by joining a domain or a workgroup. Joining a domain involves creating an account for the appliance in the given Active Directory domain. The account name can be a maximum of 15 characters, and must be unique to other names registered within the Active Directory domain. Otherwise, conflicts may occur with similarly named appliances and cause issues with functionality. After the computer account has been established, the appliance can securely query the database for information about users, groups, and shares.

Joining a workgroup implicitly leaves an Active Directory domain, and “SMB” on page 202 clients who are stored in the Active Directory database will be unable to connect to shares.

If a Kerberos realm is configured to support Kerberized NFS, the system cannot be configured to join an Active Directory domain.

Active Directory LDAP Signing

There is no configuration option for LDAP signing, as that option is negotiated automatically when communicating with a domain controller. LDAP signing operates on communication between the storage appliance and the domain controller, whereas SMB signing operations on communication between SMB clients and the storage appliance.

Active Directory Windows Server 2012 Support

Windows Server 2012 is fully supported in software version 2011.1.5 and later.

Active Directory Windows Server 2008 Support

TABLE 11-35 Active Directory 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.
	2009.Q2 4.0 - 2011.1.1	Must apply hotfix for KB951191 and apply hotfix for KB957441 as needed, see Sections A and B.
	2011.1.2 and later	Must apply hotfix for KB951191, see Section A.
Windows Server 2008 SP2	2009.Q2 4.0 - 2011.1.1	See Section C.
	2011.1.2 and later	None
Windows Server 2008 R2	2009.Q2 4.0 - 2011.1.1	See Section C.
	2011.1.2 and later	None

Active Directory Windows Server 2008 Support Section A: Kerberos issue (KB951191)

- If you upgrade to 2009.Q2.4.0 or later and your Windows 2008 domain controller is running Windows Server 2008 SP2 or R2, no action is required.
- If you upgrade to 2009.Q2.4.0 or later and your Windows 2008 domain controller is running Windows Server 2008 SP1, you must apply the hotfix described in KB951191 or install Windows 2008 SP2.

Active Directory Windows Server 2008 Support Section B: NTLMv2 issue (KB957441)

- The following applies only if your appliance is running a software version prior to 2011.1.2:
- If your Domain Controller is running Windows Server 2008 SP1 you should also apply the hotfix for `http://support.microsoft.com/kb/957441/ (http://support.microsoft.com/kb/957441/)` which resolves an NTLMv2 issue that prevents the appliance from joining the domain with its default LMCompatibilityLevel setting.
- If the LMCompatibilityLevel on the Windows 2008 SP1 domain controller is set to 5, this hot fix must be installed. After applying the hotfix you must create and set a new registry key as described in KB957441.

- If you upgrade to 2011.1.2 or later, you do not need the hotfix mentioned above.

Active Directory Windows Server 2008 Support Section C: Note on NTLMv2

- The following applies only if your appliance is running a software version prior to 2011.1.2: If your Domain Controller is running Windows Server 2008 SP2 or R2 you do not need to apply the hotfix but you must apply the registry setting as described in KB957441.
- If you upgrade to 2011.1.2 or later, no action is required.

Configuring Active Directory Using the BUI

▼ Joining a Domain

1. **Configure an Active Directory site in the “SMB” on page 202 context. (optional)**

2. **Configure a preferred domain controller in the “SMB” on page 202 context. (optional)**

3. **Enable “NTP” on page 258, or ensure that the clocks of the appliance and domain controller are synchronized to within five minutes.**

4. **Ensure that your “DNS” on page 254 infrastructure correctly delegates to the Active Directory domain, or add your domain contoller's IP address as an additional name server in the “DNS” on page 254 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.**

Configuring Active Directory Using the CLI

To demonstrate the CLI interface, the following example views the existing configuration, joins a workgroup, and then joins a domain.

Example - Configuring Active Directory Using the CLI

1. **View an existing configuration.**

```
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
```

2. **Observe that the appliance is currently operating in the domain "eng.fishworks.com". Following is an example of leaving that domain and joining a workgroup.**

```
twofish:configuration services ad> workgroup
twofish:configuration services ad workgroup> set workgroup=WORKGROUP
twofish:configuration services ad workgroup> commit
twofish:configuration services ad workgroup> done
twofish:configuration services ad> show
Properties:
                     <status> = disabled
                         mode = workgroup
                    workgroup = WORKGROUP
```

3. **Following is an example of configuring the site and preferred domain controller in preparation for joining another domain.**

```
twofish:configuration services ad> done
twofish:> configuration services smb
twofish:configuration services smb> set ads_site=sf
twofish:configuration services smb> set pdc=192.168.3.21
twofish:configuration services smb> commit
twofish:configuration services smb> show
Properties:
                     <status> = online
```

```
                 lmauth_level = 4
                          pdc = 192.168.3.21
                     ads_site = sf
twofish:configuration services smb> done
```

4. **Following is an example of joining the new domain after the properties are configured. When joining an AD domain, you must set the user and password each time you commit the node.**

```
twofish:> configuration services ad
twofish:configuration services ad> domain
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
```

Identity Mapping Service

The identity mapping service manages Windows and Unix user identities simultaneously by using both traditional Unix UIDs (and GIDs) and Windows SIDs. For information about using the BUI and CLI with Identity Mapping, see “Configuring Services Using the BUI” on page 189 and “Configuring Services Using the CLI” on page 192.

Identity Mapping Properties

The identity mapping service creates and maintains a database of mappings between SIDs, UIDs, and GIDs. Three different mapping approaches are available, if mappings are available for a given identity, the service creates an ephemeral mapping. The following mapping modes are available:

Identity Mapping Rule-based Mapping

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

Identity Mapping Directory-based Mapping

Directory-based mapping involves annotating an “LDAP” on page 238 or “Active Directory” on page 242 object with information about how the identity maps to an equivalent identity on the opposite platform. The following attributes must be assigned when using directory-based mapping:

- AD Attribute - Unix User Name - The name in the AD database of the equivalent Unix user name
- AD Attribute - Unix Group Name - The name in the AD database of the equivalent Unix group name
- Native LDAP Attribute - Windows User Name - The name in the LDAP database of the equivalent Windows identity

The CLI property names are shorter versions of those listed above.

For information on augmenting the “Active Directory” on page 242 or the “LDAP” on page 238 schemas, see the Managing Directory-Based Identity Mapping for Users and Groups (Task Map) section in the Solaris CIFS Administration Guide.

Identity Mapping IDMU

Microsoft offers a feature called "Identity Management for Unix", or IDMU. This software is available for Windows Server 2003, and is bundled with Windows Server 2003 R2 and later. This feature is part of what was called "Services For Unix" in its unbundled form.

The primary use of IDMU is to support Windows as a NIS/NFS server. IDMU adds a "UNIX Attributes" panel to the Active Directory Users and Computers user interface that lets the administrator specify a number of UNIX-related parameters: UID, GID, login shell, home directory, and similar for groups. These parameters are made available through AD through a schema similar to (but not the same as) RFC2307, and through the NIS service.

When the IDMU mapping mode is selected, the identity mapping service consumes these Unix attributes to establish mappings between Windows and Unix identities. This approach is very similar to directory-based mapping, only the identity mapping service queries the property schema established by the IDMU software instead of allowing a custom schema. When this approach is used, no other directory-based mapping may take place.

Identity Mapping Rules

This page lets you create mappings using the following properties:

- Mapping Type - Allows or denies credentials. For more information, see “Deny Mappings” on page 247.

- Mapping Direction - The mapping direction. A mapping may map credentials in both directions, only from Windows to Unix, or only from Unix to Windows. For more information, see “Mapping Rule Directional Symbols” on page 247.
- Windows Domain - The Active Directory domain of the Windows identity.
- Windows Identity - The name of the Windows identity.
- Unix Identity- The name of the Unix identity.
- Unix Identity Type - The type of the Unix identity, either a user or a group.

Deny 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” on page 202 shares for all Unix users in the group "guest". You cannot create deny mappings that conflict with other mappings.

Mapping Rule Directional Symbols

After creating a name-based mapping, the following symbols indicate the semantics of each rule.

- align="center"|↔ - Maps Windows identity to Unix identity, and Unix identity to Windows identity
- align="center"|○→ - Maps Windows identity to Unix identity
- align="center"|←○ - Maps Unix identity to Windows identity
- align="center"|○⊣ - Prevents Windows identity from obtaining credentials
- align="center"|⊢○ - Prevents Unix identity from obtaining credentials

If an icon is gray instead of black (↔, ○→, ←○, ○⊣, ⊢○), that rule matches a Unix identity which cannot be resolved.

Identity Mapping Mappings

The Mappings page 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. This page lets you view and delete exiting mappings using the Show and Flush buttons.

Identity Mapping Logs

This page shows a log of recent activity.

Identity Mapping Best Practices

- Configuring fine-grained identity mapping rules only applies when you want to have the same user access a common set of files as both an “NFS” on page 195 and “SMB” on page 202 client. If “NFS” on page 195 and “SMB” on page 202 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” on page 202 sessions. Connected users remain connected, and their previous name mapping is available for authorizing access to additional shares for up to 10 minutes. To prevent unauthorized access you must configure the mappings before you export shares.
- The security that your identity mappings provide is only as good as their synchronization with your directory services. For example, if you create a name-based mapping that denies access to a particular user, and the user's name changes, the mapping no longer denies access to that user.
- You can only have one bidirectional mapping for each Windows domain that maps all users in the Windows domain to all Unix identities. If you want to create multiple domain-wide rules, be sure to specify that those rules map *only* from Windows to Unix.
- Use the IDMU mapping mode instead of directory-based mapping whenever possible.

Identity Mapping Concepts

The “SMB” on page 202 service uses the identity mapping service to associate Windows and Unix identities. When the “SMB” on page 202 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” on page 202 and “NFS” on page 195 clients. By associating Windows and Unix identities, an “NFS” on page 195 and “SMB” on page 202 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” on page 242 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” on page 236 and “LDAP” on page 238. 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.

Identity Mapping 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” on page 202 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.

Identity 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.

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” on page 202 see that the file is owned by that Windows identity. However, “NFS” on page 195 clients see that the file is owned by "nobody".

Identity Mapping Examples

This is an 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)
```

Configuring Identity Mapping

▼ Configuring Identity Mapping

1. **Ensure that you are joined to at least one active directory domain. For information about active directories, see the "Active Directory" on page 242 section.**

2. **On the Configuration => Services => Identity Mapping => Properties page, select the Mapping mode you want to use. For information about mapping modes, see "Properties" on page 247.**

3. **If you select Directory-based Mapping, you must configure additional properties. For more information about these properties, see "Directory-based Mapping" on page 247.**

4. **To save your settings, click Apply or to start over click Revert.**

5. **To create a mappings, click Rules.**

6. **On the Rules page, click the add ⊕ icon.**

7. **In the Add Mapping Rule box, enter the required information. For more information, see "Rules" on page 247**

8. **To save your settings, click Add or click Cancel. When you create a mapping it appears in the Rules list.**

▼ Viewing or Flushing Mappings

1. **To view existing mappings, on the Configuration => Services => Identity Mapping => Mappings page, enter the required information. For information about Mappings, see "Mappings" on page 247.**

2. **Click Show. The mapping you designated appears.**

3. **To delete the mapping, click Flush. The mapping is removed.**

DNS Service

The DNS (Domain Name Service) client provides the ability to resolve IP addresses to hostnames and vice versa, and is always enabled on the appliance. Optionally, secondary hostname resolution via NIS and/or LDAP, if configured and enabled, may be requested for hostnames and addresses that cannot be resolved using DNS. Hostname resolution is used throughout the appliance user interfaces, including in "Logs" in "Oracle ZFS Storage Appliance Customer Service Manual " logs to indicate the location from which a user performed an auditable action and in "Analytics" in "Oracle ZFS Storage Appliance Analytics Guide " 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.

DNS Properties

TABLE 11-36 DNS 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 "Configuring Services Using the BUI" on page 189 and "Configuring Services Using the CLI" on page 192. The CLI property names are shorter versions of those listed above.

Configuring DNS

The CLI includes built-ins for `nslookup` and `getent hosts`, which can be used to test that hostname resolution is working:

```
caji:> nslookup deimos
192.168.1.109   deimos.sf.fishworks.com
caji:> getent hosts deimos
192.168.1.109   deimos.sf.fishworks.com
```

DNS Logs

TABLE 11-37 DNS Logs

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

Active Directory and DNS

If you plan to use "Active Directory" on page 242, the servers must be able to resolve hostname and server records in the Active Directory portion of the domain namespace. For example, if your appliance resides in the domain example.com and the Active Directory portion of the namespace is redmond.example.com, your nameservers must be able to reach an authoritative server for example.com, and they must provide delegation for the domain redmond.example.com to one or more Active Directory servers serving that domain. These are requirements imposed by Active Directory, not the appliance itself. If they are not satisfied, you will be unable to join an Active Directory domain.

Non-DNS Resolution

DNS is a standard, enterprise-grade, highly-scalable and reliable mechanism for mapping between hostnames and IP addresses. Use of working DNS servers is a best practice and will generally yield the best results. In some environments, there may be a subset of hosts that can be resolved only in NIS or LDAP maps. If this is the case in your environment, enable non-

DNS host resolution and configure the appropriate directory service(s). If LDAP is used for host resolution, the hosts map must be located at the standard DN in your database: ou=Hosts, (Base DN), and must use the standard schema. When this mode is used with NFS sharing by netgroups, it may be necessary for client systems to use the same hostname resolution mechanism configured on the appliance, or NFS sharing exceptions may not work correctly.

When non-DNS host resolution is enabled, DNS will still be used. Only if an address or hostname cannot be resolved using DNS will NIS (if enabled) and then LDAP (if enabled) be used to resolve the name or address. This can have confusing and seemingly inconsistent results. You can validate host resolution results using the `getent` CLI command described above.

Use of these options is strongly discouraged.

DNS-Less Operation

DNS-less operation is not supported on the appliance and could cause undesirable results. Several features do not operate correctly without DNS, including but not limited to:

- “Analytics” in “Oracle ZFS Storage Appliance Analytics Guide ” will be unable to resolve client addresses to hostnames.
- The “Active Directory” on page 242 feature will not function (you will be unable to join a domain).
- Use of SSL-protected “LDAP” on page 238 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.

Dynamic Routing Service

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. RIP and RIPng listen on UDP ports 520 and 521 respectively.

Dynamic Routing Logs

TABLE 11-38 Dynamic Routing

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

IPMP Service

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 Chapter 4, “Network Configuration”.

IPMP Properties

TABLE 11-39 IPMP 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 “Configuring Services Using the BUI” on page 189 and “Configuring Services Using the CLI” on page 192. The CLI property names are shorter versions of those listed above.

IPMP Logs

TABLE 11-40 IPMP Logs

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

NTP Service

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.

NTP Properties

TABLE 11-41 NTP 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: SUN7000

Changing services properties is documented in "Configuring Services Using the BUI" on page 189 and "Configuring Services Using the CLI" on page 192. The CLI property names are shorter versions of those listed above.

NTP 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.

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:

TABLE 11-42 NTP Private Keys and Integers

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.

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.

NTP 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.

TABLE 11-43 NTP Clock Synchronization

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

Configuring NTP Using the 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.

▼ 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.**

Configuring NTP Using the 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>
```

Phone Home Service

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 the `Oracle Auto Service Request (ASR) (http://oracle.com/asr)` feature. Oracle ASR automatically opens Service Requests (SR) for specific problems reported by your appliance. Registration also connects your appliance with My Oracle Support (MOS) to detect update notifications.

- The Phone Home service communicates with Oracle support to provide:
- Fault reporting - the system reports active problems to Oracle for automated service response. Depending on the nature of the fault, a support case may be opened. Details of these events can be viewed in “Problems” in “Oracle ZFS Storage Appliance Customer Service Manual ”.
- Heartbeats - daily heartbeat messages are sent to Oracle to indicate that the system is up and running. Oracle support may notify the technical contact for an account when one of the activated systems fails to send a heartbeat for too long.

- System configuration - periodic messages are sent to Oracle describing current software and hardware versions and configuration as well as storage configuration. No user data or metadata is transmitted in these messages.
- Support bundles - the Phone Home service must be enabled before support bundles can be uploaded to Oracle Support. See "System" in "Oracle ZFS Storage Appliance Customer Service Manual " for more information.
- Update Notifications - creates an Alert when new software updates are available on My Oracle Support (MOS). See "Software Update Notification" in "Oracle ZFS Storage Appliance Customer Service Manual " for more information.

You must register to use the Phone Home service.

Oracle Single Sign-On Account

You need a valid Oracle Single Sign-On account user name and password to use the fault reporting and heartbeat features of the Phone Home service. Go to `http://support.oracle.com (http://support.oracle.com)` and click Register to create your account.

Phone Home Properties

Changing services properties is documented in "Configuring Services Using the BUI" on page 189 and "Configuring Services Using the CLI" on page 192. The Phone Home service is known as `scrk` within the CLI.

Phone Home 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 Oracle. These proxy settings will also be used to upload support bundles. See "System" in "Oracle ZFS Storage Appliance Customer Service Manual " for more details on support bundles.

TABLE 11-44 Phone Home Web Proxy Settings

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

Property	Description
Password	Web proxy password

Registering the Appliance

To register the appliance for the first time, you must provide an Oracle Single Sign-On Account. Go to My Oracle Support (http://support.oracle.com) and click Register to create your account.

Registering the Appliance Using the BUI

1. **Enter your Oracle Single Sign-On Account user name and password. A privacy statement is displayed. It can be viewed at any time in both the BUI and CLI.**
2. **Commit your changes.**
3. **Use My Oracle Support (http://support.oracle.com/) to complete Auto Service Request (ASR) (http://oracle.com/asr) activation. Refer to "How To Manage and Approve Pending ASR Assets In My Oracle Support" (Doc ID 1329200.1).**

Registering the Appliance Using the CLI

1. **Set `soa_id` and `soa_password` to the user name and password for your Oracle Single Sign-On Account, respectively.**
2. **Commit your changes.**
3. **Use My Oracle Support (http://support.oracle.com/) to complete Auto Service Request (ASR) (http://oracle.com/asr) activation. Refer to "How To Manage and Approve Pending ASR Assets In My Oracle Support" (Doc ID 1329200.1).**

Example 11-1 CLI Registration

```
dory:> configuration services scrk
dory:configuration services scrk>set soa_id=myuser
                          soa_id = myuser(uncommitted)
dory:configuration services scrk> set soa_password=mypass
                   soa_password = ****** (uncommitted)
dory:configuration services scrk> commit
```

▼ Changing Account Information

1. **Click 'Change account...' to change the Oracle Single Sign-On Account used by the appliance.**
2. **Commit your changes.**
3. **Use My Oracle Support to complete Auto Service Request (ASR) activation. Refer to "How To Manage and Approve Pending ASR Assets In My Oracle Support" (Doc ID 1329200.1)**

Phone Home Status

TABLE 11-45 Phone Home Status

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

Phone Home State

If the Phone Home service is enabled before a valid Oracle Single Sign-On account has been entered, it will appear in the maintenance state. You must enter a valid Oracle Single Sign-On account to use the Phone Home service.

Phone Home Logs

There is a log of Phone Home events in “Logs” in “Oracle ZFS Storage Appliance Customer Service Manual ”.

REST

RESTful API

The ZFSSA RESTful API lets you manage the ZFSSA using simple requests such as GET, PUT, POST, and DELETE HTTP against managed resource URL paths.

The ZFSSA RESTful based architecture is defined as a layered client-server model. Advantages of this model mean that services can be transparently redirected through standard hubs, routers, and other network systems without client configuration. This architecture supports caching of information and is useful when many clients request the same static resources.

For complete ZFSSA RESTful API documentation, see Oracle ZFS Storage Appliance Documentation.

Service Tags

Service Tags are used to facilitate product inventory and support, by allowing the appliance to be queried for data such as:

- System serial number
- System type
- Software version numbers

You can register the service tags with Oracle support, allowing you to easily keep track of your Oracle equipment and also expedite service calls. The service tags are enabled by default.

Service Tag Properties

TABLE 11-46 UDP/TCP Port 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 "Configuring Services Using the BUI" on page 189 and "Configuring Services Using the CLI" on page 192. The CLI property names are shorter versions of those listed above.

SMTP Service

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.

SMTP Properties

TABLE 11-47 SMTP 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 "Configuring Services Using the BUI" on page 189 and "Configuring Services Using the CLI" on page 192. The CLI property names are shorter versions of those listed above.

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.

SMTP Logs

TABLE 11-48 SMTP Logs

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

SNMP Service

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

- Appliance status information can be served by SNMP.
- Chapter 9, "Alert Configuration" can be configured to send SNMP traps.

SNMP versions v1, v2c, and v3 are available when this service is enabled. The appliance supports a maximum of 50 physical and logical network interfaces. More than 50 network interfaces could cause time outs for such commands as snmpwalk and snmpget. If you need more than 50 network interfaces, contact Oracle Support.

SNMP Properties

- Version: Toggles between v1/2c and v3.
- Community name: Toggles between public and user-input. If you select user-input, you must also enter a community name. If you select v3, this property is not available.
- Authorized network/subnet: Enter an appropriate IPv4 address and subnet (integers from 0-32). If you select v3, this property is not available.
- Appliance contact: Enter an appropriate appliance contact.
- Username/password: Enter a valid username (max 501 characters) and password (8-501 characters). If you select v1/2c, this property is not available.
- Authentication: Toggles between MD5 and SHA authentication algorithms. If you select v1/2c, this property is not available.
- Privacy: Toggles between None and DES encryption algorithm. If you select v1/2c, this property is not available.
- Engine ID: The EngineID value hashed by snmpd. If SNMP was not previously enabled, the label shows “0x000”.
- Trap destinations: Lets you add IPv4 addresses. Use the “+” and “-” buttons to add or remove addresses.

Changing services properties is documented in “Configuring Services Using the BUI” on page 189 and “Configuring Services Using the CLI” on page 192. The CLI property names are shorter versions of those listed above.

The SNMP service also provides the MIB-II location string. This property is sourced from the “System Identity” on page 275 configuration.

SNMP MIBs

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

TABLE 11-49 SNMP MIBs

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

MIB	Purpose
.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)

Note: Sun MIB files are available at https://*your IP address or host name*:215/docs/snmp/

Sun FM MIB

The Sun FM MIB (SUN-FM-MIB.mib) provides access to SUN Fault Manager information such as:

- Active problems on the system
- Fault Manager events
- Fault Manager configuration information

There are four main tables to read:

TABLE 11-50 Sun FM MIBs

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:

TABLE 11-51 Sun AK MIBs

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.

Confinguring SNMP

▼ 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.**

5. **You must configure alerts to send the traps you want to receive. For more information about alerts, see Chapter 9, “Alert Configuration”.**

Syslog Service

The Syslog Relay service provides two different functions on the appliance:

- Chapter 9, “Alert Configuration” 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.

Syslog Properties

TABLE 11-52 Syslog Properties

Property	Description
Protocol Version	The version of the Syslog protocol to use, either Classic or Modern

Property	Description
Destinations	The list of destination IPv4 and IPv6 addresses to which messages are relayed.

Changing services properties is documented in "Configuring Services Using the BUI" on page 189 and "Configuring Services Using the CLI" on page 192. The CLI property names are shorter versions of those listed above.

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" on page 275 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" on page 275 configuration.

SYSLOG Message Format

The Syslog protocol itself does not define the format of the message payload, leaving it up to the sender to include any kind of structured data or unstructured human-readable string that is appropriate. Sun Storage appliances use the syslog subsystem tag ak to indicate a structured, parseable message payload, described next. Other subsystem tags indicate arbitrary human-readable text, but administrators should consider these string forms *unstable* and subject to change without notice or removal in future releases of the Sun Storage software.

TABLE 11-53 SYSLOG Message Formats

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" on page 247 service for POSIX and Windows identity conversion.
daemon	smbd	"SMB Data Protocol" on page 202 for accessing shares.

SYSLOG Alert Message Format

If an alert is configured with the Send Syslog Message action, it will produce a syslog message payload containing localized text consisting of the following standard fields. Each field will be prefixed with the field name in CAPITAL letters followed by a colon and whitespace character.

TABLE 11-54 SYSLOG Alert Message Formats

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.

Field Name	Description
EVENT-TIME	The time corresponding to this event. The time will be in the form "Day Mon DD HH:MM:SS YYYY" in UTC. For example: Fri Aug 14 21:34:22 2009.
PLATFORM	The platform identifier for the appliance. This field is for Oracle Service use only.
CSN	The chassis serial number of the appliance.
HOSTNAME	The canonical name of the appliance as defined by the "System Identity" on page 275 configuration.
SOURCE	The subsystem within the appliance software that emitted the event. This field is for Oracle Service use only.
REV	The internal revision of the subsystem. This field is for Oracle Service use only.
EVENT-ID	The Universally Unique Identifier (UUID) associated with this event. Oracle's Fault Management system associates a UUID with each alert and fault diagnosis such that administrators can gather and correlated multiple messages associated with a single condition, and detect duplicate messages. Oracle Service personnel can use the EVENT-ID to retrieve additional postmortem information associated with the problem that may help Oracle respond to the issue.
DESC	Description of the condition associated with the event.
AUTO-RESPONSE	The automated response to the problem, if any, by the Fault Management software included in the system. Automated responses include capabilities such as proactively offlining faulty disks, DRAM memory chips, and processor cores.
REC-ACTION	The recommended service action. This will include a brief summary of the recommended action, but administrators should consult the knowledge article and this documentation for information on the complete repair procedure.

The SEVERITY field will be set to one of the following values:

TABLE 11-55 SYSLOG Severity Fields

Severity	Syslog Level	Description
Minor	LOG_WARNING	A condition occurred that does not currently impair service, but the condition needs to be corrected before it becomes more severe.

Severity	Syslog Level	Description
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 immediate 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.
```

System Identity

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

System Identity Properties

TABLE 11-56 System Identity 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

Property	Description
System Location	A text string to describe the where the appliance is physically located. If "SNMP" on page 266 is enabled, this will be exported as the *syslocation* string in MIB-II

Changing services properties is documented in "Configuring Services Using the BUI" on page 189 and "Configuring Services Using the CLI" on page 192. The CLI property names are shorter versions of those listed above.

System Identity Logs

TABLE 11-57 System Identity Logs

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

SSH Service

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" in "Oracle ZFS Storage Appliance Analytics Guide " statistics.

SSH Properties

TABLE 11-58 SSH 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 “Configuring Services Using the BUI” on page 189 and “Configuring Services Using the CLI” on page 192. The CLI property names are shorter versions of those listed above.

SSH Logs

TABLE 11-59 SSH Logs

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

Configuring SSH

▼ Disabling root SSH access

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

♦ ♦ ♦ CHAPTER 12

Shares, Projects, and Schema

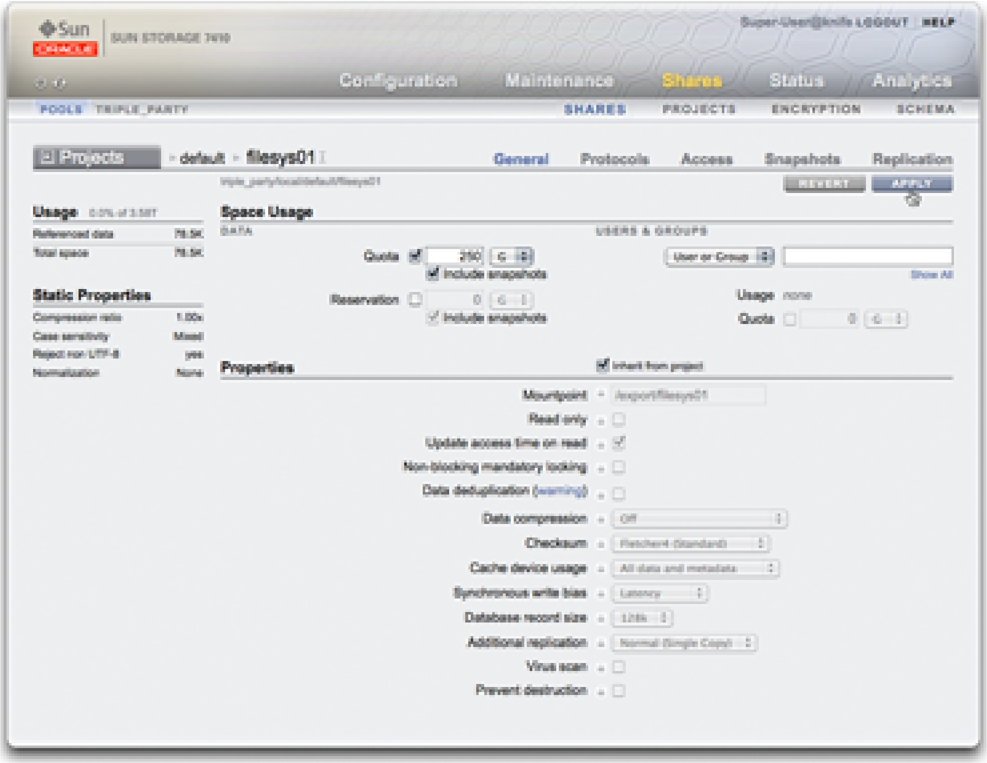
This section describes ZFSSA shares, projects and schema.

For common administrative purposes, including space management and common settings, shares can be grouped into ZFSSA projects. In addition to the standard built in properties, you can configure any number of additional properties that are available on all shares and projects. These properties are given basic types for validation purposes, and are inherited like most other standard properties. The values are never consumed by the software in any way, and exist solely for end-user consumption. The property schema is global to the system, across all pools, and is synchronized between cluster peers.

Understanding Shares

Storage Pools

FIGURE 12-1 Similar Shares can be Grouped in a Project.



triple_party/local/default/filesys01

Usage 0.0% of 3.58T

Referenced data	78.5K
Total space	78.5K

Static Properties

Compression ratio	1.00x
Case sensitivity	Mixed
Reject non UTF-8	yes
Normalization	None

Space Usage

DATA

Quota ☑ 250 G
☑ Include snapshots

Reservation □ 0 G
☑ Include snapshots

USERS & GROUPS

User or Group

Usage none

Quota □ 0 G

Properties ☑ Inherit from project

Mountpoint	/export/filesys01
Read only	□
Update access time on read	☑
Non-blocking mandatory locking	□
Data deduplication (warning)	□
Data compression	Off
Checksum	Fletcher4 (Standard)
Cache device usage	All data and metadata
Synchronous write bias	Latency
Database record size	128k
Additional replication	Normal (Single Copy)
Virus scan	□
Prevent destruction	□

The ZFSSA 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 Chapter 5, “Storage Configuration” on the ZFSSA. Once a storage pool is configured, there is no need to statically size filesystems, though this behavior can be achieved by using “Shares Space Management” on page 283.

While multiple storage pools are supported, this type of configuration is generally discouraged because it provides significant drawbacks as described in Chapter 5, “Storage Configuration”. 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.

Using Shares

Shares are filesystems and LUNs that are exported over supported data protocols to clients of the ZFSSA. Filesystems export a file-based hierarchy and can be accessed over “SMB” on page 202, “NFS” on page 195, “HTTP/WebDav” on page 219, and “FTP” on page 217. LUNs export block-based volumes and can be accessed over “iSCSI” on page 200 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.

Share Properties

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

TABLE 12-1 Project and Share Properties

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 mount point and SMB properties. When inherited, these properties concatenate the project setting with their own share name.
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.

Property Type	Description
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 “Schemas” on page 346.

Share 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” on page 326 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.

Share Clones

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

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.

Shares Space Management

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” on page 280 page, the ZFSSA 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.

Shares Space Terminology

Before getting into details, it is important to understand some basic terms used when talking about space usage on the ZFSSA.

- 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 ZFSSA 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 Z will not free up the space, but destroying both X and Y will. Because of this, destroying any snapshot can affect the unique space referenced by neighboring snapshots, though the total amount of space consumed by snapshots will always decrease.

The total size of a project or share always accounts for space consumed by all snapshots, though the usage breakdown is also available. Quotas and reservations can be set at the project level to enforce physical constraints across this total space. In addition, quotas and reservations can be set at the filesystem level, and these settings can apply to only referenced data or total data. Whether or not quotas and reservations should be applied to referenced data or total physical data depends on the administrative environment. If users are not in control of their snapshots (i.e. an automatic snapshot schedule is set for them), then quotas should typically not include snapshots in the calculation. Otherwise, the user may run out of space but be confused when files cannot be deleted. Without an understanding of snapshots or means to manage those snapshots, it is possible for such a situation to be unrecoverable without administrator intervention. In this scenario, the snapshots represent an overhead cost that is factored into operation of the system in order to provide backup capabilities. On the other hand, there are environments where users are billed according to their physical space requirements, and snapshots represent a choice by the user to provide some level of backup that meets their requirements given the churn rate of their dataset. In these environments, it makes more sense to enforce quotas based on total physical data, including snapshots. The users understand the cost of snapshots, and can be provided a means to actively management them (as through dedicated roles on the ZFSSA).

File System and Project Settings

The simplest way of enforcing quotas and reservations is on a per-project or per-filesystem basis. Quotas and reservations do not apply to LUNs, though their usage is accounted for in the total project quota or reservations.

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" on page 283, 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" on page 283, 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.

Space Management for Replicating LUNs

When you create a LUN the full physical space you configure for the LUN is reserved and cannot be used by other file systems (unless it is thinly provisioned). For replication, when you take a snapshot of a LUN of any given size, up to twice the size of the LUN is also reserved, depending on how much of the LUN space has been used.

The following list shows the maximum overhead space required when replicating a LUN:

- Up to 100% on the source between updates
- Up to 200% on the source during an update
- Up to 200% on the target

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 “Updates” in “Oracle ZFS Storage Appliance Customer Service Manual ” 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.

▼ Viewing Current Usage in the BUI

1. **To view the current usage in the BUI, go to Shares > Shares > General.**

2. **In the Space Usage - Users and Groups section click the User or Group drop down to select User or Group and query the current usage for any given user or group within a share or across a project.**

3. **Type the name of the User or Group that you want to query. The query progresses as you type.**

When the lookup is completed, the current usage is displayed. In addition, the "Show All" link shows a dialog with a list of current usage of all users or groups. This dialog can only query for a particular type - users or groups - and does not support querying both at the same time. This list displays the canonical UNIX and Windows name (if mappings are enabled), as well as the usage and (for filesystems) quota.

Viewing Current Usage in the CLI

1. **In the CLI, use the `users` and `groups` commands in the context of a particular project or share.**

2. **Use, the `show` command to display current usage in a tabular form.**

3. **To retrieve the usage for a particular user or group, select the user or group you want and use the `get` command.**

```
clownfish:> shares select default
clownfish:shares default> users
clownfish:shares default users> list
USER       NAME                           USAGE
user-000   root                            325K
user-001   ahl                            9.94K
user-002   eschrock                       20.0G
clownfish:shares default users> select name=eschrock
clownfish:shares default user-002> get
                          name = eschrock
                      unixname = eschrock
                        unixid = 132651
                       winname = (unset)
                         winid = (unset)
                         usage = 20.0G
```

Setting User or Group Quotas

Quotas can be set on a user or group at the filesystem level. 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.

▼ Set User or Group Quotas Using the BUI

1. **In the BUI, go to Shares > Shares > General.**

2. **In the Space Usage - Users and Groups section click the User or Group drop down to select User or Group and query the current usage for any given user or group within a share or across a project.**

3. **In the browser, user quotas are managed from the “general” on page 304 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.**

4. **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.**

▼ Set User or Group Quotas Using the 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” on page 247 service on the ZFSSA. 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

Every filesystem on the ZFSSA must be given a unique mountpoint which serves as the access point for the filesystem data. Projects can be given mountpoints, but these serve only as a tool to manage the namespace using inherited properties. Projects are never mounted, and do not export data over any protocol.

All shares must be mounted under `/export`. While it is possible to create a filesystem mounted at `/export`, it is not required. If such a share doesn't exist, any directories will be created dynamically as necessary underneath this portion of the hierarchy. Each mountpoint must be unique within a cluster.

Namespace Nested Mountpoints

It is possible to create filesystems with mountpoints beneath that of other filesystems. In this scenario, the parent filesystems are mounted before children and vice versa. The following cases should be considered when using nested mountpoints:

- If the mountpoint doesn't exist, one will be created, owned by root and mode 0755. This mountpoint may or may not be torn down when the filesystem is renamed, destroyed, or moved, depending on circumstances. To be safe, mountpoints should be created within the parent share before creating the child filesystem.
- If the parent directory is read-only, and the mountpoint doesn't exist, the filesystem mount will fail. This can happen synchronously when creating a filesystem, but can also happen asynchronously when making a large-scale change, such as renaming filesystems with inherited mountpoints.
- When renaming a filesystem or changing its mountpoint, all children beneath the current mountpoint as well as the new mountpoint (if different) will be unmounted and remounted after applying the change. This will interrupt any data services currently accessing the share.
- Support for automatically traversing nested mountpoints depends on protocol, as outlined below.

Namespace 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.

Namespace 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.

Namespace 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.

Namespace SMB

The SMB protocol does not use mountpoints, as each share is made available by resource name. However, each filesystem must still have a unique mountpoint. Nested mountpoints (multiple filesystems within one resource) are not currently supported, and any attempt to traverse a mountpoint will result in an empty directory.

Namespace FTP / FTPS / SFTP

Filesystems are exported using their standard mountpoint. Nested mountpoints are fully supported and are transparent to the user. However, it is not possible to not share a nested filesystem when its parent is shared. If a parent mountpoint is shared, then all children will be shared as well.

Namespace HTTP / HTTPS

Filesystems are exported under the `/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 > Shares

Working with Shares > Shares in the 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:

TABLE 12-2 BUI Shares List

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" on page 294 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.

Field	Description
GUID	The SCSI GUID for the LUN. See "Shares > Shares > Protocols - BUI Page" on page 311 for more information.

The following tools are available for each share:

TABLE 12-3 BUI Shares > Shares Icons

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" on page 304
- "Protocols" on page 311
- "Access" on page 318
- "Snapshots" on page 326
- Chapter 13, "Replication"

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. The following usage statistics are shown:

- Available space - This statistic is implicitly shown as the capacity in terms of capacity percentage in the title. The available space reflects any quotas on the share or project, or the absolute capacity of the pool. The number shown here is the sum of the total space used and the amount of available space.
- Referenced Data - The amount of data referenced by the data. This includes all filesystem data or LUN blocks, in addition to requisite metadata. With compression, this value may be much less than the logical size of the data contained within the share. If the share is a clone of a snapshot, this value may be less than the physical storage it could theoretically include, and may be zero.
- Snapshot Data - The amount of space used by all snapshots of the share, including any project snapshots. This size is not equal to the sum of unique space consumed by all snapshots. Blocks that are referenced by multiple snapshots are not included in the per-snapshot usage statistics, but will show up in the share's snapshot data total.
- Unused Reservation - If a filesystem has a reservation set, this value indicates the amount of remaining space that is reserved for the filesystem. This value is not set for LUNs. The ZFSSA 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” on page 304 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. The following static properties are shown:

- 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” on page 304 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,

BUI Value	CLI Value	Description
		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 1M, and the default is 8K.
- Origin - If this is a clone, this is the name of the snapshot from which it was cloned.
- Data Migration Source - If set, then this filesystem is actively shadowing an existing filesystem, either locally or over NFS. For more information about data migration, see the section on Chapter 14, “Shadow Migration”.

Shares Project Panel

In the BUI, the set of available projects is always available via the project panel at the left side of the view. To expand or collapse the project panel, click the triangle by the "Projects" title bar.

TABLE 12-4 Project Panel Icons

Icon	Description
▢	Expand project panel

Icon	Description
▶	Collapse project panel

Selecting a project from the panel will navigate to the “Projects” on page 335 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” on page 335 view.

▼ Creating a Share

1. **To view shares in a project or across all projects, go to Shares > Shares.**

2. **Select Filesystems or LUNs.**

3. **Click the plus icon next to Filesystems or Luns.**

   The Create Filesystem or Create LUN dialog box appears.

4. **In the Create Filesystem or Create LUN dialog box, select or type the properties you want to use.**

   The properties for each type of shares are defined in the following locations:

   For Filesystems:

   - “User” on page 318
   - “Group” on page 318
   - “Permissions” on page 318
   - “Mountpoint” on page 304
   - Chapter 12, “Shares, Projects, and Schema” (create time only)
   - Chapter 12, “Shares, Projects, and Schema” (create time only)
   - Chapter 12, “Shares, Projects, and Schema” (create time only)

   For LUNs:

   - “Volume size” on page 304
   - “Thin provisioned” on page 304

- Chapter 12, “Shares, Projects, and Schema” (create time only)

Working with Shares > Shares in the 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 = discard (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)
                     snaplabel = project1:share1
                      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 = discard (inherited)
                         atime = true (inherited)
                      checksum = fletcher4 (inherited)
                   compression = off (inherited)
                        copies = 1 (inherited)
                    mountpoint = /export/foo (inherited)
                         quota = 0 (inherited)
                      readonly = false (inherited)
                    recordsize = 128K (inherited)
                   reservation = 0 (inherited)
                secondarycache = all (inherited)
                        nbmand = false (inherited)
                      sharesmb = off (inherited)
                      sharenfs = on (inherited)
                       snapdir = hidden (inherited)
                     snaplabel = project1:share1
                         vscan = false (inherited)
                      sharedav = off (inherited)
                      shareftp = off (inherited)
                    root_group = other (default)
              root_permissions = 700 (default)
                     root_user = nobody (default)
               casesensitivity = (default)
                 normalization = (default)
                      utf8only = (default)
                    quota_snap = (default)
              reservation_snap = (default)
                    custom:int = (default)
                 custom:string = (default)
                  custom:email = (default)
clownfish:shares default/foo (uncommitted)> set 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" on page 283 section.

Shares > Shares CLI Properties

The following properties are available in the CLI, with their equivalent in the BUI. Properties can be set using the standard CLI commands get and set. In addition, properties can be inherited from the parent project by using the unset command.

TABLE 12-5 Shares > Shares CLI Properties

CLI Name	"Type" on page 280	BUI Name	BUI Location
aclinherit	inherited	"ACL inheritance behavior" on page 318	Access
aclmode	inherited	"ACL behavior on mode change" on page 318	Access
atime	inherited	"Update access time on read" on page 304	General
casesensitivity	create time	Chapter 12, "Shares, Projects, and Schema"	Static
checksum	inherited	Chapter 12, "Shares, Projects, and Schema"	General
compression	inherited	"Data compression" on page 304	General
compresratio	read-only	Chapter 12, "Shares, Projects, and Schema"	Static
copies	inherited	"Additional replication" on page 304	General
creation	read-only	-	-
dedup	inherited	"Data deduplication" on page 304	General

CLI Name	**“Type” on page 280**	BUI Name	BUI Location
exported	inherited, replication packages only	Chapter 13, “Replication”	General
fixednumber	LUN local	“Initiator group” on page 311	Protocols
initiatorgroup	LUN local	“Initiator group” on page 311	Protocols
logbias	inherited	“Synchronous write bias” on page 304	General
lunumber	LUN local	“LU number” on page 311	Protocols
lunguid	read-only, LUN local	“GUID” on page 311	Protocols
mountpoint	inherited	“Mountpoint” on page 304	General
nbmand	inherited	“Non-blocking mandatory locking” on page 304	General
nodestroy	inherited	“Prevent destruction” on page 304	General
normalization	create time	Chapter 12, “Shares, Projects, and Schema”	Static
origin	read-only	Chapter 12, “Shares, Projects, and Schema”	Static
quota	space management	“Quota” on page 283	General
quota_snap	space management	“Quota / Include snapshots” on page 283	General
readonly	inherited	“Read-only” on page 304	General
recordsize	inherited	“Database record size” on page 304	General
reservation	space management	“Reservation” on page 283	General
reservation_snap	space management	“Reservation / Include snapshots” on page 283	General
root_group	filesystem local	“Group” on page 318	Access
root_permissions	filesystem local	“Permissions” on page 318	Access
root_user	filesystem local	“User” on page 318	Access
rstchown	inherited	“Restrict ownership change” on page 304	General

CLI Name	“Type” on page 280	BUI Name	BUI Location
secondary cache	inherited	“Cache device usage” on page 304	General
shadow	create time	Chapter 14, “Shadow Migration”	Static
sharedav	inherited	“Protocols / HTTP / Share mdoe” on page 311	Protocols
shareftp	inherited	“Protocols / FTP / Share mode” on page 311	Protocols
sharenfs	inherited	“Protocols / NFS / Share mode” on page 311	Protocols
sharesmb	inherited	“Protocols / SMB / Resource name” on page 311	Protocols
snapdir	inherited	“.zfs/snapshot visibility” on page 326	Snapshots
snaplabel	inherited	“Scheduled snapshot label” on page 326	Snapshots
space_available	read-only	Chapter 12, “Shares, Projects, and Schema”	Usage
space_data	read-only	Chapter 12, “Shares, Projects, and Schema”	Usage
space_snapshots	read-only	Chapter 12, “Shares, Projects, and Schema”	Usage
space_total	read-only	Chapter 12, “Shares, Projects, and Schema”	Usage
space_unused_res	read-only	Chapter 12, “Shares, Projects, and Schema”	Usage
sparse	LUN local	“Thin provisioned” on page 304	General
targetgroup	LUN local	“Target group” on page 311	Protocols
utf8only	create time	“Reject non UTF-8”	Static
volblocksize	create time	Chapter 12, “Shares, Projects, and Schema”	Static
vscan	inherited	“Virus scan” on page 304	General

Shares > Shares > General - BUI Page

This section of the BUI controls overall settings for the share that are independent of any particular protocol and are not related to access control or snapshots. While the CLI groups all properties in a single list, this section describes the behavior of the properties in both contexts.

These are standard properties that can either be inherited from the project or explicitly set on the share. The BUI only allows the properties to be inherited all at once, while the CLI allows for individual properties to be inherited.

For information on how these properties map to the CLI, see the “Working with Shares > Shares in the CLI” on page 299 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 managing space usage for filesystems, including quotas and reservations, see the “Space Management” on page 283 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” on page 304 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 "Project - Reservation" on page 342.

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 ZFSSA.

Mountpoints can be nested underneath other shares, though this has some limitations. For more information, see the "filesystem namespace" on page 291 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" on page 318 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 Chapter 8, “Setting ZFSSA Preferences” and then use “Analytics” in “Oracle ZFS Storage Appliance Analytics Guide ” 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 ZFSSA, 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

BUI value	CLI value	Description
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 Chapter 5, “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 disjointed filesystems, this can cause contention on the log devices that can increase latency across all consumers. Even with multiple filesystems requesting synchronous semantics, it may be the case that some filesystems are more latency-sensitive than others.

A common case is a database that has a separate log. The log is extremely latency sensitive, and while the database itself also requires synchronous semantics, it is heavier bandwidth and not latency sensitive. In this environment, setting this property to 'throughput' on the main database while leaving the log filesystem as 'latency' can result in significant performance improvements. This setting will change behavior even when no log devices are present, though the effects may be less dramatic.

The Synchronous write bias setting can be bypassed by the Oracle Intelligent Storage Protocol. Instead of using the write bias defined in the file system, the Oracle Intelligent Storage Protocol can use the write bias value provided by the Oracle Database NFSv4 client. The write bias value sent by the Oracle Database NFSv4 client is used only for that write request. For more information, see “ Oracle Intelligent Storage Protocol ” on page 463.

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

Specifies a suggested block size for files in the file system. This property is only valid for filesystems and is designed for use with database workloads that access files in fixed-size records. The system automatically tunes block sizes according to internal algorithms optimized for typical access patterns.

For databases that create very large files but access them in small random chunks, these algorithms may be suboptimal. Specifying a record size greater than or equal to the record size of the database can result in significant performance gains. Use of this property for general purpose file systems is strongly discouraged, and may adversely affect performance.

The default record size is 128 KB. The size specified must be a power of two greater than or equal to 512 and less than or equal to 1 MB. Changing the file system's record size affects only files created afterward; existing files and received data are unaffected.

NOTE: If block sizes greater than 128K are used for projects or shares, replication of those projects or shares to systems that don't support large block sizes will fail.

The Database record size setting can be bypassed by the Oracle Intelligent Storage Protocol. Instead of using the record size defined in the file system the Oracle Intelligent Storage Protocol can use the block size value provided by the Oracle Database NFSv4 client. The block size provided by the Oracle Database NFSv4 client can only be applied when creating a new database files or table. Block sizes of existing files and tables will not be changed. For more information, see “ Oracle Intelligent Storage Protocol ” on page 463.

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” on page 233 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 ZFSSA 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 "Schemas" on page 346.

Shares > Shares > Protocols - BUI Page

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" on page 195, "SMB" on page 202, "HTTP" on page 219, and "FTP" on page 217 properties apply only to filesystems, while the "iSCSI" on page 200 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:

TABLE 12-6 Share Protocols

Protocol	Location
NFS	twofish:/export/fs0
SMB	\\twofish\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 Chapter 6, “Storage Area Network Configuration”.

Share Protocols - NFS

TABLE 12-7 Share Protocols - NFS Properties

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:

TABLE 12-8 Client Collection 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" on page 236 (if enabled) and then from "LDAP" on page 238 (if enabled). If LDAP is used, the netgroups must be found at the default location, ou=Netgroup,(Base DN), and must use the standard schema. The username component of a netgroup entry typically has no effect on NFS; only the hostname is significant. Hostnames contained in netgroups must be canonical and, if resolved using DNS, fully qualified. That is, the NFS subsystem will attempt to verify that the IP address of the requesting client resolves to a canonical hostname that matches either the specified FQDN or one of the members of one of the specified netgroups. This match must be exact, including any domain components; otherwise, the exception will not match and the next exception will be tried. For more information on hostname resolution, see "DNS" on page 254. Management of netgroups can be complex; consider using IP subnet rules or DNS domain rules instead where possible.

As of the 2013.1.0 software release, Unix client users may belong to a maximum of 1024 groups without any performance degradation. Prior releases supported up to 16 groups per Unix client user.

Share Protocols - CLI

In the CLI, all NFS share modes and exceptions are specified using a single options string for the "sharenfs" property. This string is a comma-separated list of values from the tables above. It should begin with one of "ro", "rw", or "off", as an analogue to the global share modes described for the BUI. For example,

```
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.

TABLE 12-9 Kerberos Security Settings

Setting	Description
krb5	End-user authentication through Kerberos V5
krb5i	krb5 plus integrity protection (data packets are tamper proof)

Setting	Description
krb5p	krb5i plus privacy protection (data packets are tamper proof and encrypted)

Combinations of Kerberos flavors may be specified in the security mode setting. The combination security modes let clients mount with any Kerberos flavor listed.

TABLE 12-10 Security Mode Settings

Setting	Menu
sys	System Authentication
krb5	Kerberos v5 only - Clients must mount using this flavor.
krb5:krb5i	Kerberos v5, with integrity - Clients may mount using any flavor listed.
krb5i	Kerberos v5 integrity only - Clients must mount using this flavor.
krb5:krb5i:krb5p	Kerberos v5, with integrity or privacy - Clients may mount using any flavor listed.
krb5p	Kerberos v5 privacy only - Clients must mount using this flavor.

For more information about NFS and Kerberos, see:

- `http://www.ietf.org/rfc/rfc2623.txt (http://www.ietf.org/rfc/rfc2623.txt)` (NFSv2 and NFSv3 Security)
- `http://www.ietf.org/rfc/rfc3530.txt (http://www.ietf.org/rfc/rfc3530.txt)` (NFSv4 Protocol)

Character Set Encodings

Normally, the character set encoding used for filename is unspecified. The NFSv3 and NFSv2 protocols don't specify the character set. NFSv4 is supposed to use UTF-8, but not all clients do and this restriction is not enforced by the server. If the UTF-8 only option is disabled for a share, these filenames are written verbatim to the filesystem without any knowledge of their encoding. This means that they can only be interpreted by clients using the same encoding. SMB, however, requires filenames to be stored as UTF-8 so that they can be interpreted on the server side. This makes it impossible to support arbitrary client encodings while still permitting access over SMB.

In order to support such configurations, the character set encoding can be set share-wide or on a per-client basis. The following character set encodings are supported:

- cp932
- euc-cn
- euc-jp
- euc-jpms
- euc-kr
- euc-tw
- iso8859-1
- iso8859-2
- iso8859-5
- iso8859-6
- iso8859-7
- iso8859-8
- iso8859-9
- iso8859-13
- iso8859-15
- koi8-r
- shift_jis

The default behavior is to leave the character set encoding unspecified (pass-through). The BUI allows the character set to be chosen through the standard exception list mechanism. In the CLI, each character set itself becomes an option with one or more hosts, with '*' indicating the share-wide setting. For example, the following:

```
set sharenfs="rw,euc-kr=*"
```

Will share the filesystem with 'euc-kr' as the default encoding. The following:

```
set sharenfs="rw,euc-kr=host1.domain.com,euc-jp=host2.domain.com"
```

Use the default encoding for all clients except 'host1' and 'host2', which will use 'euc-kr' and 'euc-jp', respectively. The format of the host lists follows that of other CLI NFS options.

Note that some NFS clients do not correctly support alternate locales; consult your NFS client documentation for details.

Shares - SMB

- Resource name - The name by which “SMB” on page 202 clients refer to this share. The resource name "off" indicates no “SMB” on page 202 client may access the share, and the resource name "on" indicates the share will be exported with the filesystem's name.
- Enable Access-based Enumeration - An option which, when enabled, performs access-based enumeration. Access-based enumeration filters directory entries based on the

credentials of the client. When the client does not have access to a file or directory, that file will be omitted from the list of entries returned to the client. This option is not enabled by default.

- Is a DFS Namespace - A property which indicates whether this share is provisioned as a standalone "DFS namespace" on page 202.
- 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" on page 202 shares on the same system may share the same resource name. Resource names inherited from projects have special behavior, see "Projects" on page 335 for details. Resource names must be less than 80 characters, and can contain any alphanumeric characters besides the following characters:

```
" / \ [ ] : | < > + ; , ? * =
```

When access-based enumeration is enabled, clients may see directory entries for files which they cannot open. Directory entries are filtered only when the client has no access to that file. For example, if a client attempts to open a file for read/write access but the ACL grants only read access, that open request will fail but that file will still be included in the list of entries.

Shares - iSCSI

- Target group - The targets over which this LUN is exported.
- Initiator group(s) - The initiators that can access this LUN. As of the 2013.1.0 software release, multiple initiator groups can be assigned to a LUN. When editing initiator groups, checking the PERSIST checkbox (the default) preserves the LUN number for the corresponding initiator group. If unchecked, the ZFSSA may reassign the LUNs after a SAN configuration change or a reboot.
- LU (logical unit) number - As LUNs are associated with target and initiator groups, they are assigned unique logical unit numbers per target group and initiator pair. No two LUNs that are accessible by an initiator through a target group may share a logical unit number. This property controls whether a logical unit must have number zero or an automatically assigned number.
- Operational status - The operational status of this LUN. An offline LUN is inaccessible to initiators regardless of target or initiator configuration.
- Write cache behavior - This setting controls whether the LUN caches writes. With this setting off, all writes are synchronous and if no log device is available, write performance suffers significantly. Turning this setting on can therefore dramatically improve write performance, but can also result in data corruption on unexpected shutdown or failover

unless the client application understands the semantics of a volatile write cache and properly flushes the cache when necessary. Consult your client application documentation before turning this on.

- GUID - A LUN's GUID is a globally unique, read-only identifier that identifies the SCSI device. This GUID remains consistent within different head nodes and replicated environments.

Shares - HTTP

TABLE 12-11 Shares - HTTP Properties

Property	Description
Share mode	The HTTP share mode for this filesystem. One of none, read only, or read/write.

Shares - FTP

TABLE 12-12 Shares - FTP Properties

Property	Description
Share mode	The FTP share mode for this filesystem. One of none, read only, or read/write.

Shares - SFTP

TABLE 12-13 Shares - SFTP Properties

Property	Description
Share mode	The SFTP share mode for this filesystem. One of none, read only, or read/write.

Shares > Shares > Access

Access Control

This view lets you set options to control ACL behavior as well as control access to the root directory of the filesystem. This view is only available for filesystems.

Shares - 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.

Shares - 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" on page 247 service. For Unix-based NFS access, this can be changed from the client using the `chown` command.

Shares - 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" on page 247 service. For Unix-based NFS access, this can be changed from the client using the `chgrp` command.

Shares - Permissions

Standard Unix permissions for the root directory. For Unix-based NFS access, this can be changed from the client using the `chmod` command. The permissions are divided into three types.

TABLE 12-14 Shares Users

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.

TABLE 12-15 Shares Permissions

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.

- As of the 2011.1 software release, the following additional behavior is associated with the "write" permission for all directories:
- Child files within the directory can be deleted (same as the ACL D permission), unless the sticky bit is set on the directory, in which case the child files can be deleted only if requested by the owner of the file
- Times associated with a file or directory can be changed (same as the ACL A permission)
- Extended attributes can be created, and writes are allowed to the extended attributes directory (same as the ACL W permission)

In the BUI, selecting permissions is done by clicking on individual boxes. Alternatively, clicking on the label ("user," "group," or "other) will select (or deselect) all permissions within the label. In the CLI, permissions are specified as a standard Unix octal value, where each digit corresponds to (in order) user, group, and other. Each digit is the sum of read (4), write (2), and execute (1). So a permissions value of 743 would be the equivalent of user RWX, group R, other WX.

As an alternative to setting POSIX permission bits at share creation time, administrators may instead select the "Use Windows Default Permissions" option, which will apply an ACL as described in the “root directory ACL” on page 318 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.

Shares - ACL Behavior

For information on ACLs and how they work, see the “root directory ACL” on page 318 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.

TABLE 12-16 Mode Change Values

BUI Value	CLI Value	Description
Discard ACL	discard	All ACL entries that do not represent the mode of the directory or file are discarded. This is the default behavior.
Mask ACL with mode	mask	The permissions are reduced, such that they are no greater than the group permission bits, unless it is a user entry that has the same UID as the owner of the file or directory. In this case, the ACL permissions are reduced so that they are no greater than owner permission bits. The mask value also preserves the ACL across mode changes, provided an explicit ACL set operation has not been performed.
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 usually only affect ACL entries that are flagged as inheritable - other entries are not propagated regardless of this property setting. However, all trivial ACL entries are inheritable when used with SMB. A trivial ACL represents the traditional Unix `owner/group/other` entries.

TABLE 12-17 ACL Inheritance Behavior Values

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.

BUI Value	CLI Value	Description
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.

When using SMB to create a file in a directory with a trivial ACL, all ACL entries are inherited. As a result, the following behavior occurs:

- Inheritance bits display differently when viewed in SMB or NFS. When viewing the ACL directory in SMB, inheritance bits are displayed. In NFS, inheritance bits are not displayed.
- When a file is created in a directory using SMB, its ACL entries are shown as inherited; however, when viewed through NFS, the directory has no inheritable ACL entries.
- If the ACL is changed so that it is no longer trivial, e.g., by adding an access control entry (ACE), this behavior does not occur.
- If the ACL is modified using SMB, the resulting ACL will have the previously synthetic inheritance bits turned into real inheritance bits.

All of the above behavior is subject to change in a future release.

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 ZFSSA supports NFSv4-style ACLs, also accessible over SMB. POSIX draft ACLs (used by NFSv3) are not supported. Some trivial ACLs can be represented over NFSv3, but making complicated ACL changes may result in undefined behavior when accessed over NFSv3.

Like root directory access, this property only affects the root directory of the filesystem. ACLs can be controlled through in-band protocol management, but the BUI provides a way to set the ACL just for the root directory of the filesystem. There is no way to set the root directory ACL through the CLI. You can use in-band management tools if the BUI is not an option. Changing this ACL does not affect existing files and directories in the filesystem. Depending on the ACL inheritance behavior, these settings may or may not be inherited by newly created files and directories. However, all ACL entries are inherited when SMB is used to create a file in a directory with a trivial ACL.

An ACL is composed of any number of ACEs (access control entries). Each ACE describes a type/target, a mode, a set of permissions, and inheritance flags. ACEs are applied in order, starting at the beginning of the ACL, to determine whether a given action should be permitted. For information on in-band configuration ACLs through data protocols, consult the appropriate client documentation. The BUI interface for managing ACLs and the effect on the root directory are described here.

TABLE 12-18 Share - ACL Types

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.

TABLE 12-19 Share - ACL Modes

Mode	Description
○ Allow	The permissions are explicitly granted to the ACE target.

Mode	Description
Deny	The permissions are explicitly denied to the ACE target.

TABLE 12-20 Share - ACL Permissions

	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.
(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.
(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.
(d)	Delete	Permission to delete a file.
(D)	Delete Child	Permission to delete a file within a directory. As of the 2011.1 software release, if the sticky bit is set, a child file can only be deleted by the file owner.

	Permission	Description
(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.
(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:

TABLE 12-21 Share Root Directory Entities

Type	Action	Access
Owner	Allow	Full Control
Group	Allow	Read and Execute
Everyone	Allow	Read and Execute

Shares - Snapshots

Snapshots are read only copies of a filesystem at a given point of time. For more information on snapshots and how they work, see the “concepts” on page 280 page.

Shares - Snapshot Properties

.zfs/snapshot visible

Filesystem snapshots can be accessed over data protocols at `.zfs/snapshot` in the root of the filesystem. This directory contains a list of all snapshots on the filesystem, and they can be accessed just like normal filesystem data (in read only mode). By default, the '.zfs' directory is not visible when listing directory contents, but can be accessed by explicitly looking it up. This prevents backup software from inadvertently backing up snapshots in addition to new data.

TABLE 12-22 Snapshot Values

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.

Scheduled Snapshot Label

This optional property appends a user-defined label to each scheduled snapshot and is blank by default. The label can either be set for an individual share, or set for a project and inherited by its shares, but not both. Snapshot labels can help identify the project or share for which a snapshot was taken, for example "project1:share1" could indicate a scheduled snapshot taken on share1 within project1. Labels can be up to 35 alphanumeric characters and include special characters _ - . :

Listing Snapshots Using the BUI

Under the "snapshots" tab is the list of active snapshots of the share. This list is divided into two tabs: the "Snapshots" tab is used for browsing and managing snapshots. The "Schedules" tab manages automatic snapshot schedules. Within the "Snapshots" tab, you can select between viewing all snapshots, only manual snapshots, or only scheduled snapshots. For each snapshot, the following fields are shown:

Field	Description
Name	The name of the snapshot. There are two types of snapshots: manual and automatic.
	Manual Snapshots: "Name" is the name provided when the snapshot was created. Manual snapshots can be renamed by clicking on the name and entering a new value.
	Automatic Snapshots: There are three types, and they cannot be renamed:
	- .auto: User-configured scheduled snapshots with custom retention policies (see “Scheduled Snapshots” on page 326).
	- .ndmp: Used for NDMP backup and automatically pruned.
	- .rr: Used for remote replication and automatically pruned.
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” on page 280 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.

Manual Snapshots Using the BUI

There are two types of snapshots; project level and share/LUN level snapshots.

Create a project level snapshot

1. **Open the project that you want to snapshot.**
2. **Click the Snapshots tab.**
3. **Click the ⊕ icon. The snapshots list appears.**
4. **In the dialog box, type a name for the snapsnot.**
5. **To create the snapshot, click "apply".**

Create a share/LUN level snapshot

1. **Open the share/LUN that you want to snapshot.**
2. **Click the Snapshots tab.**
3. **Click the ⊕ icon. The snapshots list appears.**
4. **In the dialog box, type a name for the snapsnot.**
5. **To create the snapshot, click "apply".**

   There is no limit on the number of snapshots that can be taken, but each snapshot consumes memory, so creating large numbers of snapshots can slow the system. The practical limit on the number of snapshots system-wide depends on the system configuration, but should be over a hundred thousand.

Renaming a Snapshot (BUI)

1. **To rename a snapshot, click the name within the list of active snapshots. This will change to a text input box.**
2. **After updating the name within the text input, hitting return or changing focus will commit the changes.**

Destroying a Snapshot (BUI)

1. **To destroy a snapshot, click the 🗑 icon when over the row for the target snapshot.**

2. **Destroying a snapshot will require destroying any clones and their descendents. If this is the case, you will be prompted with a list of the clones that will be affected.**

Rolling back to a Snapshot (BUI)

1. **To rollback a filesystem, click the ↺ icon for the destination snapshot.**

2. **A confirmation dialog will appear, and if there are any clones of the snapshot, any newer snapshots, or their descendents, they will be displayed, indicating that they will be destroyed as part of this process.**

   In addition to accessing the data in a filesystem snapshot directory, snapshots can also be used to roll back to a previous instance of the filesystem or LUN. This requires destroying any newer snapshots and their clones, and reverts the share contents to what they were at the time the snapshot was taken. It does not affect any property settings on the share, though changes to filesystem root directory access will be lost, as that is part of the filesystem data.

Cloning a Snapshot (BUI)

- **To create a clone, click the ⊞ icon for the source snapshot. A dialog will prompt for the following values.**

  - Project - Destination project. By default, clones are created within the current project, but they can also be created in different projects (or moved between projects).
  - Name - Type a name for the clone.
  - Mountpoint - To use this value, click the lock icon. Set the mountpoint for the clone. When Retain Other Local Settings is set, the clone must be given a different mountpoint, as shares cannot save the same mountpoint.
  - Resource Name - To use this value, click the lock icon. Enter the resource that you want to use for the clone.
  - Retain Other Local Settings - By default, all currently inherited properties of the filesystem will inherit from the destination project in the clone. Local settings are always preserved. Setting this property causes any inherited properties to be preserved as local setting in the new clone.

A “clone” on page 280 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.

Scheduled Snapshots Using the BUI

In addition to manual snapshots, you can configure automatic snapshots according to the table below. These snapshots are named ".auto-<timestamp>", and can be taken on half hourly, hourly, daily, weekly, or monthly schedules. A schedule is a list of intervals and retention policies.

Times are displayed in the local (client browser) time zone. However, times are stored and executed in UTC format and without regard to such conventions as daylight saving time. For example, a snapshot scheduled for 10:00 a.m. PST (UTC-8) is stored and executed at 18:00 UTC.

Automatic snapshots can be set on a project or a share, but not both. Otherwise, overlapping schedules and retention policies would make it impossible to guarantee both schedules. Removing an interval, or changing its retention policy, will immediately destroy any automatic snapshots not covered by the new schedule. Automatic snapshots with clones are ignored.

Previous versions of the software allowed for automatic snapshots at the frequency of a minute. This proved to put undue strain on the system and was not generally useful. To help users avoid placing undue stress on the system, this feature was removed with the 2010.Q3 release. Snapshots can now only be specified at a period of once every half hour or longer. Existing minute periods will be preserved should the software be rolled back, and previous instances will expire according to the existing schedule, but no new snapshots will be taken. An alert will be posted if a share or project with this frequency is found.

- **To add a new interval, click the ⊕ icon when viewing the "Schedules" tab. Each interval has the following properties.**

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

Property	Description
	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 ZFSSA is down for a day, when you come back up all your hour snapshots will be deleted. Snapshots that are part of multiple intervals are only destroyed when no interval specifies that they should be retained.

Manual Snapshots Using the CLI

To access share snapshots, navigate to the share and the snapshots context.

```
clownfish:> shares select default select builds
clownfish:shares default/builds> snapshots
clownfish:shares default/builds snapshots>
```

Listing Snapshots (CLI)

Snapshots can be listed using the standard CLI commands.

```
clownfish:shares default/builds snapshots> list
today
yesterday
clownfish:shares default/builds snapshots>
```

Taking Manual Snapshots (CLI)

To take a manual project-level snapshot, navigate to the project and snapshot node and then use the `snapshot` command:

```
clownfish:cd /
clownfish:shares select myproject snapshots
clownfish:shares myproject snapshots> snapshot cob_monday
```

To take a manual share-level snapshot of an individual share, navigate to that share and use the snapshot command there:

```
clownfish:cd /
clownfish:shares select myproject select share1 snapshots
clownfish:snapshot lunchtime
```

Renaming a Snapshot (CLI)

To rename a snapshot, use the `rename` command:

```
clownfish:shares default/builds snapshots> rename test test2
clownfish:shares default/builds snapshots>
```

Destroying a Snapshot (CLI)

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 (CLI)

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 (CLI)

To clone a snapshot, use the `clone` command. This command will place you into an uncommitted share context identical to the one used to create shares. From here, you can adjust properties as needed before committing the changes to create the clone.

```
clownfish:shares default/builds snapshots> select today
clownfish:shares default/builds@today> clone testbed
clownfish:shares default/testbed (uncommitted clone)> get
                    aclinherit = restricted (inherited)
                       aclmode = discard (inherited)
                         atime = true (inherited)
                      checksum = fletcher4 (inherited)
                   compression = off (inherited)
                        copies = 1 (inherited)
                    mountpoint = /export/testbed (inherited)
                         quota = 0 (default)
                      readonly = false (inherited)
                    recordsize = 128K (inherited)
                   reservation = 0 (default)
                secondarycache = all (inherited)
                        nbmand = false (inherited)
                      sharesmb = off (inherited)
                      sharenfs = on (inherited)
                       snapdir = hidden (inherited)
                         vscan = false (inherited)
                      sharedav = off (inherited)
                      shareftp = off (inherited)
                    root_group = other (default)
              root_permissions = 777 (default)
                     root_user = nobody (default)
                    quota_snap = true (default)
              reservation_snap = true (default)
clownfish:shares default/testbed (uncommitted clone)> set quota=10G
                         quota = 10G (uncommitted)
clownfish:shares default/testbed (uncommitted clone)> commit
clownfish:shares default/builds@today>
```

The command also supports an optional first argument, which is the project in which to create the clone. By default, the clone is created in the same project as the share being cloned.

Listing Dependent Clones Using the CLI

To list all clones created from a particular snapshot (dependent clones), navigate to the snapshot and then use the list clones command.

```
clonefish:shares default/builds> snapshots
clonefish:shares default/builds snapshots> select today
```

```
clonefish:shares default/builds@today> list clones

Clones: 2 total

PROJECT          SHARE
default          testbed
default          production
clonefish:shares default/builds@today>
```

The result shows clone names and the project where the clone resides.

Scheduled Snapshots Using the CLI

Automatic scheduled snapshots can be configured using the `automatic` command from the snapshot context, at the project level of that of an individual share. Once in this context, new intervals can be added and removed with the `create` and `destroy` commands. Each interval has a set of properties that map to the BUI view of the frequency, offset, and number of snapshots to keep. Schedules are maintained in UTC format.

```
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>
```

Setting the Scheduled Snapshot Label Using the CLI

In the BUI, the "scheduled snapshot label" property can be set for either a project or a share. Likewise, in the CLI, the label can be set by first navigating to either the project or share context. To create a scheduled snapshot label, use the `set snaplabel` command:

```
clownfish:shares project1/share1> set snaplabel=project1:share1
```

Projects

Shares, filesystems and LUNs can be grouped into projects. A project defines a common administrative control point for managing shares. shares within a project can share common settings, and quotas can be enforced at the project level in addition to the share level. Projects can also be used solely for grouping logically related shares together, so their common attributes (such as accumulated space) can be accessed from a single point.

By default, the ZFSSA 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.

Working with Projects Using the 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.

Project Fields

After navigating to the project view, you will be presented with a list of projects on the system. Alternatively, you can navigate to the shares screen and open the project panel for a shortcut to projects. The panel does not scale well to large numbers of projects, and is not a replacement for the complete project list. The following fields are displayed for each project:

TABLE 12-23 Project Fields

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:

TABLE 12-24 Project Icons

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 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 “Shares > Shares” on page 293.
- Referenced data - Sum of all referenced data for all shares within the project, in addition to a small amount of project overhead. See “Shares > Shares” on page 293 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 “Shares > Shares” on page 293 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 “Shares > Shares” on page 293 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 “Shares > Shares” on page 293 for a complete description.

▼ Creating Projects

1. **To create a project, view the list of projects and click the ⊕ button.**
2. **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.**

Working with Projects Using the 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 = discard
                         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
```

```
                     snaplabel = project1:share1
                         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)
                     snaplabel = project1:share1
```

```
                         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 top level 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.

Project 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" on page 280	BUI Name	BUI Location
aclinherit	inherited	"Project Access" on page 344	Access
aclmode	inherited	"Project Access" on page 344	Access

CLI Name	"Type" on page 280	BUI Name	BUI Location
atime	inherited	"Project - General" on page 341	General
checksum	inherited	"Project - General" on page 341	General
compression	inherited	"Project - General" on page 341	General
compressratio	read-only	"Projects" on page 335	Static
copies	inherited	"Project - General" on page 341	General
creation	read-only	-	-
dedup	inherited	"Project - General" on page 341	General
default_group	creation default	"Project - General" on page 341	General
default_permissions	creation default	"Project - General" on page 341	General
default_sparse	creation default	"Project - General" on page 341	General
default_user	creation default	"Project - General" on page 341	General
default_volblocksize	creation default	"Project - General" on page 341	General
default_volsize	creation default	"Project - General" on page 341	General
mountpoint	inherited	"Project - General" on page 341	General
nbmand	inherited	"Project - General" on page 341	General
quota	space management	"Project - General" on page 341	General
readonly	inherited	"Project - General" on page 341	General
recordsize	inherited	"Project - General" on page 341	General
reservation	space management	"Project - General" on page 341	General
secondary cache	inherited	"Project - General" on page 341	General

CLI Name	“Type” on page 280	BUI Name	BUI Location
sharedav	inherited	“Project Protocols” on page 343	Protocols
shareftp	inherited	“Project Protocols” on page 343	Protocols
sharenfs	inherited	“Project Protocols” on page 343	Protocols
sharesmb	inherited	“Project Protocols” on page 343	Protocols
snapdir	inherited	“Project Snapshots” on page 344	Snapshots
snaplabel	inherited	“Project Snapshots” on page 344	Snapshots
space_available	read-only	“Projects” on page 335	Usage
space_data	read-only	“Projects” on page 335	Usage
space_snapshots	read-only	“Projects” on page 335	Usage
space_total	read-only	“Projects” on page 335	Usage
space_unused_res	read-only	“Projects” on page 335	Usage
space_unused_res_shares	read-only	“Projects” on page 335	Usage
vscan	inherited	“Project - General” on page 341	General

Project - General

Project - General 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.

Project - 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” on page 280 page.

Project - 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" on page 304. Unlike filesystems, project quotas cannot exclude snapshots, and can only be enforced across all shares and their snapshots.

Project - Reservation

Guarantees a minimum amount of space for use across all filesystems and LUNs within the project. For more information, see the "shares section" on page 304. Unlike filesystems, project reservation cannot exclude snapshots, and can only be enforced across all shares and their snapshots.

Project - Inherited Properties

These are standard properties that can either be inherited by shares within the project. The behavior of these properties is identical to that at the shares level, and further documentation can be found in the shares section.

- "Mountpoint" on page 304
- "Read only" on page 304
- "Update access time on read" on page 304
- "Non-blocking mandatory locking" on page 304
- "Data compression" on page 304
- "Data deduplication" on page 304
- "Checksum" on page 304
- "Cache device usage" on page 304
- "Database record size" on page 304
- "Additional replication" on page 304
- "Virus scan" on page 304

Project - Custom Properties

Custom properties can be added as needed to attach user-defined tags to projects and shares. For more information, see "Schemas" on page 346.

Filesystem Creation Defaults

These settings are used to fill in the default values when creating a filesystem. Changing them has no effect on existing filesystems. More information can be found in the appropriate shares section.

- “User” on page 318
- “Group” on page 318
- “Permissions” on page 318

LUN Creation Defaults

These settings are used to fill in the default values when creating a LUN. Changing them has no effect on existing LUNs. More information can be found in the appropriate shares section.

- “Volume size” on page 304
- “Thin provisioned” on page 304
- “Shares > Shares” on page 293

Project Protocols

Each project has protocol-specific properties which define the behavior of different protocols for that shares within that project. In general, “shares” on page 311 inherit protocol-specific properties in a straightforward manner. Exceptions and special cases are noted here.

- NFS - “NFS” on page 195 share properties are inherited normally, and described in the “shares documentation” on page 311.
- SMB
  - Resource name - The name by which “SMB” on page 202 clients refer to this share.
  - Enable Access-based Enumeration - An option which, when enabled, performs access-based enumeration. Access-based enumeration filters directory entries based on the credentials of the client. When the client does not have access to a file or directory, that file will be omitted from the list of entries returned to the client. This option is not enabled by default.

    No two “SMB” on page 202 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:
  - off - The contained filesystems are not exported over “SMB” on page 202.
  - on - The contained filesystems are exported over “SMB” on page 202 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” on page 200 properties are not inherited.
- HTTP - “HTTP” on page 219 share properties are inherited normally, and described in the “shares documentation” on page 311.
- FTP - “FTP” on page 217 share properties are inherited normally, and described in the “shares documentation” on page 311.
- SFTP - “SFTP” on page 229 share properties are inherited normally, and described in the “shares documentation” on page 311.
- NFS - “NFS” on page 195 share properties are inherited normally, and described in the “shares documentation” on page 311.
- SMB
  - Resource name - The name by which “SMB” on page 202 clients refer to this share.
  - Enable Access-based Enumeration - An option which, when enabled, performs access-based enumeration. Access-based enumeration filters directory entries based on the credentials of the client. When the client does not have access to a file or directory, that file will be omitted from the list of entries returned to the client. This option is not enabled by default.

    No two “SMB” on page 202 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:
  - Off - The contained filesystems are not exported over “SMB” on page 202.
  - On - The contained filesystems are exported over “SMB” on page 202 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” on page 200 properties are not inherited.

Project Access

- Access Control - This view provides control over inheritable properties that affect “ACL” on page 318 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” on page 318
  - “ACL inheritance behavior” on page 318

Project Snapshots

Snapshots are read only copies of a filesystem at a given point of time. For more information on snapshots and how they work, see the “concepts” on page 280 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.

Project Snapshot Properites

.zfs/snapshot visible

Filesystem snapshots can be accessed over data protocols at `.zfs/snapshot` in the root of the filesystem. This directory contains a list of all snapshots on the filesystem, and they can be accessed just like normal filesystem data (in read only mode). By default, the '.zfs' directory is not visible when listing directory contents, but can be accessed by explicitly looking it up. This prevents backup software from inadvertently backing up snapshots in addition to new data.

TABLE 12-25 Project Snapshot Values

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.

Scheduled Snapshot Label

This optional property appends a user-defined label to each scheduled snapshot and is blank by default. The label can either be set for an individual share, or set for a project and inherited by its shares, but not both. Snapshot labels can help identify the project or share for which a snapshot was taken, for example "project1:share1" could indicate a scheduled snapshot taken on share1 within project1. Labels can be up to 35 alphanumeric characters and include special characters _ - . :

Project level snapshots are administered in the same way as share level snapshots. For more information about snapshots, see “Shares:Snapshots” on page 326

Project snapshots do not support rollback or clone operations. For more information about snapshots, see “Shares:Snapshots” on page 326

To access the snapshots for a project, navigate to the project and run the `snapshots` command.

```
clownfish:> shares select default
clownfish:shares default> snapshots
clownfish:shares default snapshots>
```

From this point, snapshots are administered in the same way as share level snapshots. For more information about snapshots, see “Shares:Snapshots” on page 326

Project snapshots do not support rollback or clone operations. For more information about snapshots, see “Shares:Snapshots” on page 326

Schemas

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.

Working with Schemas in the BUI

To define custom properties, access the "Shares -> Schema" navigation item. The current schema is displayed as a list, and entries can be added or removed as needed. Each property has the following fields:

TABLE 12-26 Schema Property 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
TYPE	The property type, for validation purposes. This must be one of the types described below.

The valid types for properties are the following

TABLE 12-27 Valid Types for Properties

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" on page 304 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.

▼ Configuring a Schema Using the BUI

1. **Navigate to the "Shares -> Schema" view**
2. **Click the '+' icon to add a new property to the schema property list**
3. **Enter the name of the property ("contact")**
4. **Enter a description of the property ("Owner Contact")**
5. **Choose a type for the new property ("Email Address")**
6. **Click the "Apply" button**
7. **Navigate to an existing share or project**
8. **Change the "Owner Contact" property under the "Custom Properties" section.**

Working with Schemas Using the 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
```

▼ Configuring a Schema Using 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 13

Replication

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

Replication Overview

Oracle ZFS Storage Appliances support snapshot-based replication of projects and shares from a source ZFSSA to any number of target ZFSSAs 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 ZFSSA for disaster recovery. In the event of a disaster that impacts service of the primary ZFSSA (or even an entire data center), administrators activate service at the disaster recovery site, which takes over using the most recently replicated data. When the primary site has been restored, data changed while the disaster recovery site was in service can be migrated back to the primary site and normal service restored. Such scenarios are fully testable before such a disaster occurs.
- Data distribution. Replication can be used to distribute data (such as virtual machine images or media) to remote systems across the world in situations where clients of the target ZFSSA wouldn't ordinarily be able to reach the source ZFSSA 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 ZFSSAs 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. For a full update, the entire project contents up to the snapshot are sent. For an incremental update, only the changes since the last replication snapshot for the same action are sent.
- Block-level. Each update operation traverses the filesystem at the block level and sends the appropriate filesystem data and metadata to the target.
- Asynchronous. Because replication takes snapshots and then sends them, data is necessarily committed to stable storage before replication even begins sending it. Continuous replication effectively sends continuous streams of filesystem changes, but it's still asynchronous with respect to NAS and SAN clients.
- Includes metadata. The underlying replication stream serializes both user data and ZFS metadata, including most properties configured on the Shares screen. These properties can be modified on the target after the first replication update completes, though not all take effect until the replication connection is severed. For example, this allows sharing over NFS to a different set of hosts than on the source. See "Managing Replication Packages " on page 363 for details.
- Secure. The replication control protocol used among ZFS Storage Appliances is secured with SSL. Data can optionally be protected with SSL as well. Appliances can only replicate to/from other ZFSSAs after an initial manual authentication process, see "Creating and Editing Targets" on page 356.

Replication has the following known limitations:

- Changing a target's IP address will break the replication
- Actions cannot move between pools
- I/O is limited to a maximum of 200 MB/s per project level replication

Understanding Replication

Replication Terminology

- replication peer (or just peer, in this context): a ZFS Storage Appliance that has been configured as a replication source or target.
- replication source (or just source): an ZFSSA peer containing data to be replicated to another ZFSSA peer (the *target*). Individual ZFSSAs 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 ZFSSA peer that will receive and store data replicated from another ZFSSA peer (the *source*). This term also refers to a configuration object on the ZFSSA that enables it to replicate to another ZFSSA.
- 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 Replication " on page 355.

- replication action (or just action): a configuration object on a source ZFSSA specifying a project or share, a target ZFSSA, 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 ZFSSA that manages the data replicated as part of a particular action from a particular source. Each action on a source ZFSSA is associated with exactly one package on a target ZFSSA 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).

Project Replication Targets

Before a source ZFSSA can replicate to a target, the two systems must set up a replication peer connection that enables the ZFSSAs to identify each other securely for future communications. Administrators set up this connection by creating a new replication target on the Configuration > Services > Remote Replication screen on the source ZFSSA. To create a new target, administrators specify three fields:

- Name (used only to identify the target in the source ZFSSA's BUI and CLI)
- Network address or hostname (to contact the target ZFSSA)
- Target ZFSSA's root password (to authorize the administrator to set up the connection on the target ZFSSA)

The ZFSSAs then exchange keys used to securely identify each other in subsequent communications. These keys are stored persistently as part of the ZFSSA's configuration and persist across reboots and upgrades. They will be lost if the ZFSSA is factory reset or reinstalled. The root password is never stored persistently, so changing the root password on either ZFSSA does not require any changes to the replication configuration. The password is never transmitted in the clear because this initial identity exchange (like all replication control operations) is protected with SSL.

By default, the replication target connection is not bidirectional. If an administrator configures replication from a source A to a target B, B cannot automatically use A as a target. However, the system supports reversing the direction of replication, which automatically creates a target for A on B (if it does not already exist) so that B can replicate back to A.

NOTE: When a replication source uses NIS or LDAP services to map users or user groups and those users or user groups are included in a share configuration on the source (for example in 'Share Level ACL' or 'Share Space Usage'), those users or user groups should be available on the replication target (for example by using the same NIS or LDAP servers) otherwise replication sever/reverse operations may fail.

To configure replication targets, see "Configuring Project Replication " on page 356.

Project Replication Actions and Packages

Targets represent a connection between ZFSSAs 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 ZFSSA. Actions are the primary administrative control point for replication, each one specifying:

- a replication group (a project and some number of shares)
- a target ZFSSA
- a storage pool on the target ZFSSA (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 vs. Share-level Replication " on page 355). The target ZFSSA 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 ZFSSA. Each action corresponds to a *package* on the target ZFSSA that contains an exact copy of the source project and shares on which the action is configured as of the start time of the last replication update. Administrators configure the frequency and other options for replication updates by modifying properties of the corresponding action. Creating the action on the source ZFSSA creates the package on the target ZFSSA 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 ZFSSA. Once this initial sync completes, subsequent replication updates are *incremental*: only the changes since the previous update are sent. The action (on the source) and package (on the target) keep track of which changes have been replicated to the target through named replication snapshots. Generally, as long as at least one full sync has been sent for an action and the action/package connection has not been corrupted due to a software failure or administrative action, replication updates will be incremental.

The action and package are bound to each other. If the package is somehow corrupted or destroyed, the action will not be able to send replication updates, even if the target still has the data and snapshots associated with the action. Similarly, if the action is destroyed, the package will be unable to receive new replication updates (even if the source still has the same data and snapshots). The BUI and CLI warn administrators attempting to perform operations that would destroy the action-package connection. If an error or explicit administrative operation breaks the action-package connection such that an incremental update is no longer possible, administrators must sever or destroy the package and action and create a new action on the source.

NOTE: The ZFSSA avoids destroying data on the target unless explicitly requested by the administrator. As a result, if the initial replication update for an action fails after replicating some of the data and leaving incomplete data inside the package, subsequent replication updates using the same action will fail because the ZFSSA cannot overwrite the previously received data. To resolve this, administrators should destroy the existing action and package and create a new action and package and start replication again.

In software releases prior to 2010.Q1, action and replica configuration (like target configuration) was stored on the controller rather than as part of the project and share configuration in the storage pool. As a result, a factory reset caused configuration to be destroyed. In 2010.Q1 and later releases, the action and package configuration is stored in the storage pool with the corresponding projects and shares and will be available even after a factory reset. However, target information will still be lost, and actions with missing targets currently cannot be configured to point to a new target.

Project Replication Storage Pools

When the action is initially configured, the administrator is given a choice of which storage pool on the target should contain the replicated data. The storage pool containing an action cannot be changed once the action has been created. Creating the action creates the empty package on the target in the specified storage pool, and after this operation the source has no knowledge of the storage configuration on the target. It does not keep track of which pool the action is being replicated to, nor is it updated with storage configuration changes on the target.

When the target is a clustered system, the chosen storage pool must be one owned by same head which owns the IP address used by the source for replication because only those pools are always guaranteed to be accessible when the source contacts the target using that IP address. This is exactly analogous to the configuration of NAS clients (NFS and SMB), where the IP address and path requested in a mount operation must obey the same constraint. When performing operations that change the ownership of storage pools and IP addresses in a cluster, administrators must consider the impact to sources replicating to the cluster. There is currently no way to move packages between storage pools.

Project-level vs. Share-level Replication

The ZFSSA allows administrators to configure remote replication on both the project and share level. Like other properties configurable on the Shares screen, each share can either inherit or override the configuration of its parent project. Inheriting the configuration means not only that the share is replicated on the same schedule to the same target with the same options as its parent project is, but also that the share will be replicated in the same stream using the same project-level snapshots as other shares inheriting the project's configuration. This may be important for applications which require consistency between data stored on multiple shares. Overriding the configuration means that the share will not be replicated with any project-level actions, though it may be replicated with its own share-level actions that will include the project. It is not possible to override part of the project's replication configuration and inherit the rest.

More precisely, the replication configuration of a project and its shares define some number of replication *groups*, each of which is replicated with a single stream using snapshots taken simultaneously. All groups contain the project itself (which essentially just includes its properties). One project-level group includes all shares inheriting the replication configuration of the parent project. Any share that overrides the project's configuration forms a new group consisting of only the project and the share itself.

For example, suppose we have 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`.

Due to current limitations, do not mix project- and share-level replications within the same project. This avoids unpredictable results when reversing the replication direction or when replicating clones. For more details, see sections “Managing Replication Packages ” on page 363 and “ Replicating Clones ” on page 383.

Configuring Project Replication

Be sure to read and understand the above sections on replication targets, actions, and packages before configuring replication.

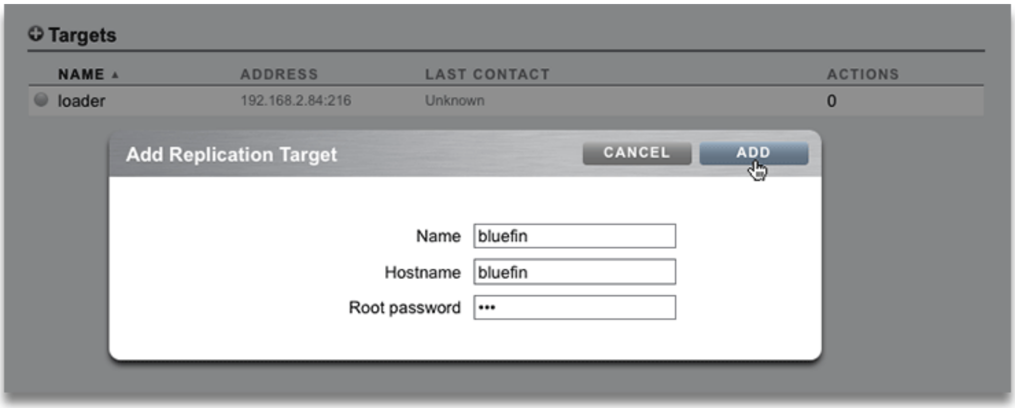
Creating and Editing Targets

This section describes creating and editing targets.

▼ Creating and Editing Targets in the BUI

1. **To create remote replication targets in the BUI, go to Configuration > Services > Remote Replication > Targets. Click ⊕ Targets and configure a Name, Hostname and Password.**

2. **To edit remote replication targets in the BUI, go to Configuration > Services > Remote Replication > Targets. For the target you want to edit, move the cursor over the target name, click the pencil icon, and configure the Name and/or Hostname. The Hostname must resolve to the same ZFSSA as before (checked by the serial number of the target). If you want to point to a different ZFSSA than previously configured, you must create a new target to authenticate against the new ZFSSA.**



NAME ▲	ADDRESS	LAST CONTACT	ACTIONS
● loader	192.168.2.84:216	Unknown	0

Name: bluefin
Hostname: bluefin
Root password: •••

▼ Creating and Editing Targets in the CLI

1. **In the CLI, navigate to the `targets` node to set or unset the target `hostname`, `root_password`, and `label` .**

```
knife:> configuration services replication targets
```

2. **From this context, administrators can:**

- Add new targets
- View the actions configured with the existing target
- Edit the unique identifier (label) and/or hostname for the target

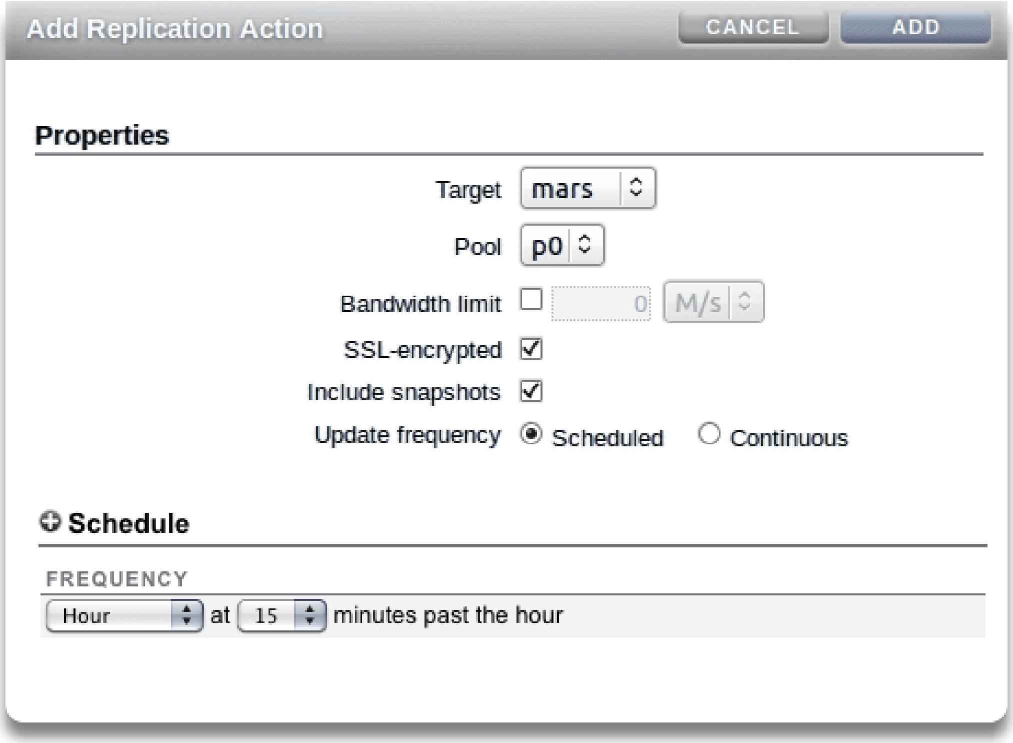
- Destroy a target, if no actions are using it

3. **A target should not be destroyed while actions are using it. Such actions will be permanently broken. The system makes a best effort to enforce this but cannot guarantee that no actions exist in exported storage pools that are using a given target.**

Creating and Editing Actions

Replication actions have the following properties, which are presented slightly differently in the BUI and CLI:

FIGURE 13-1 Add Replication Action



Properties

Target: mars

Pool: p0

Bandwidth limit: ☐ 0 M/s

SSL-encrypted: ☑

Include snapshots: ☑

Update frequency: ● Scheduled ○ Continuous

Schedule

FREQUENCY

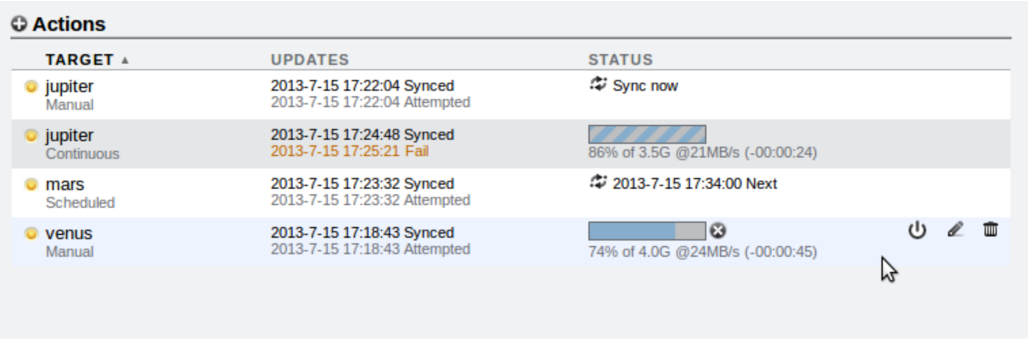
Hour at 15 minutes past the hour

TABLE 13-1 Replication Action CLI Properties

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.
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.
Mode (CLI: continuous) and schedule	Whether this action is being replicated continuously or at manual or scheduled intervals. See “Replication Modes: Scheduled or Continuous ” on page 362 for details.
Include Snapshots	Whether replication updates include non-replication snapshots. See “Replication - Including Intermediate Snapshots ” on page 362 for details.
Limit bandwidth	Specifies a maximum speed for this replication update (in terms of amount of data transferred over the network per second). Changes made to this property during a replication update do not take effect until the next update.
Bytes sent	Read-only property describing the number of bytes sent to the target.
Estimated size	Read-only property describing the estimated size of the data to be replicated.
Estimated time left	Read-only property describing the estimated time remaining until completion.
Average throughput	Read-only property describing the average replication throughput.
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."

Creating and Editing Actions in the BUI

1. **After at least one replication target has been configured, administrators can configure actions on a local project or share by navigating to it in the BUI and clicking the Replication tab or navigating to it in the CLI and selecting the "replication" node. These interfaces show the status of existing actions configured on the project or share, the replication progress information, and allow administrators to create new actions:**



TARGET ▲	UPDATES	STATUS
jupiter Manual	2013-7-15 17:22:04 Synced 2013-7-15 17:22:04 Attempted	Sync now
jupiter Continuous	2013-7-15 17:24:48 Synced 2013-7-15 17:25:21 Fail	86% of 3.5G @21MB/s (-00:00:24)
mars Scheduled	2013-7-15 17:23:32 Synced 2013-7-15 17:23:32 Attempted	2013-7-15 17:34:00 Next
venus Manual	2013-7-15 17:18:43 Synced 2013-7-15 17:18:43 Attempted	74% of 4.0G @24MB/s (-00:00:45)

2. **When replicating to a target, two rows of status information are displayed. The first row shows the target name, the date and time of the last successful synchronization, and a progress bar, or a barber-pole progress bar if the replication is continuous. The second row displays the replication type (Scheduled, Manual, or Continuous), the date and time of the last attempted or failed synchronization, and status details. For replications in progress, the status details contain the percentage of completion, the estimated size of the data to be replicated, the average replication throughput, and the estimated completion time. When replication is not in progress, the Status column displays either the next scheduled replication or the "Sync now" message, as appropriate for the replication type.**

Creating and Editing Actions in the CLI

1. **The same progress information can be displayed in the CLI, with the state `sending` shown for a replication in progress:**

```
otoro:shares otoro-proj-01 action-000> show
Properties:
                              id = 80a96f4f-93fe-4abd-eb54-fb82e7f8c69f
```

```
               target = chutoro
           continuous = false
        include_snaps = true
        max_bandwidth = unlimited
           bytes_sent = 505M
       estimated_size = 3.0G
  estimated_time_left = 00:00:41
   average_throughput = 63MB/s
              use_ssl = false
                state = sending
    state_description = Sending update
          next_update = Sync now
            last_sync = Sun Jul 14 2013 06:04:38 GMT+0000 (UTC)
             last_try = Sun Jul 14 2013 06:04:38 GMT+0000 (UTC)
          last_result = success
```

2. **Note: Replication can take a long time to complete, depending on the size of the data being replicated. Use the progress information to determine the update status. For initial replication, it is important to not interrupt it, including, but not limited to, restarting the ZFSSA or canceling the update; otherwise, the entire initial replication must be restarted.**

3. **Replication target information can be displayed in the CLI with the state `actions`:**

```
otoro:configuration services replication targets> show

Targets:
     TARGET        LABEL            ACTIONS
     target-000    oakmeal          1

otoro:configuration services replication targets> select target-000

otoro:configuration services replication target-000> show
Properties:
               address = 10.153.34.167:216
                 label = oakmeal
              hostname = oakmeal-7320-167
                   asn = 4913649f-7549-6d2a-866b-987ddbc4e163
               actions = 1

oakmeal-7320-167:configuration services replication target-000> actions
     POOL          PROJECT          SHARE
     pool1         project1         (multiple)
```

4. **When using the CLI, it can be helpful to know the ID of the newly created replication action. The ID is used later to select the correct replication action node. To view the ID of the newly created action, use the command `last` to navigate to the node with the new replication action. Then use the command `get id` to retrieve the action ID:**

```
otoro:> shares
otoro:shares> select p1
otoro:shares p1> replication
otoro:shares p1 replication> create
otoro:shares p1 action (uncommitted)> set target=oakmeal
                       target = oakmeal (uncommitted)
otoro:shares p1 action (uncommitted)> set pool=p
                         pool = p (uncommitted)
otoro:shares p1 action (uncommitted)> set use_ssl=false
                      use_ssl = false (uncommitted)
otoro:shares p1 action (uncommitted)> commit
otoro:shares p1 replication> last
otoro:shares p1 action-001> get id
                           id = fb1bb3fd-3361-42e1-e4a1-b06c426172fb
otoro:shares p1 action-001> done
otoro:shares p1 replication>
```

Replication Modes: Scheduled or Continuous

Replication actions can be configured to send updates on a schedule or continuously. The replication update process itself is the same in both cases. This property only controls the interval.

Because continuous replication actions send updates as frequently as possible, they result in sending a constant stream of all filesystem changes to the target system. For filesystems with a lot of churn (many files created and destroyed in short intervals), this can result in replicating much more data than actually necessary. However, as long as replication can keep up with data changes, this results in the minimum data lost in the event of a data-loss disaster on the source system.

Continuous replication is still asynchronous. ZFS Storage Appliances do not currently support synchronous replication, which does not consider data committed to stable storage until it's committed to stable storage on both the primary and secondary storage systems.

Replication - 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.

Replication - Sending and Canceling Updates

For targets that are configured with scheduled or manual replication, administrators can immediately send a replication update by clicking the ⇄ button in the BUI or using the `sendupdate` command in the CLI. This is not available (or will not work) if an update is actively being sent. Ensure there is enough disk space on the target to replicate the entire project before sending an update.

If an update is currently active, the BUI displays a progress bar, and the CLI shows a state of `sending`. To cancel the update, click the ⊗ button or use the `cancelupdate` command. It may take several seconds before the cancellation completes.

Managing Replication Packages

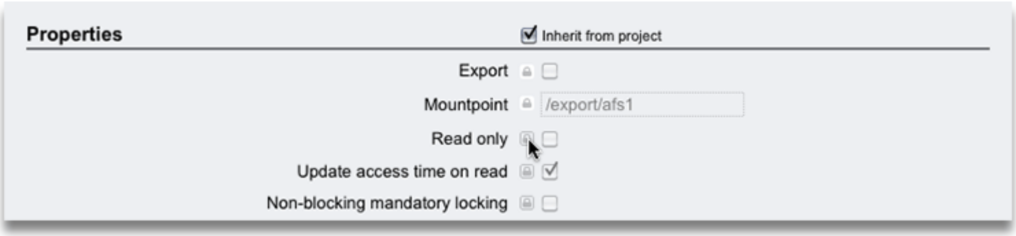
Packages are containers for replicated projects and shares. Each replication action on a source ZFSSA corresponds to one package on the target ZFSSA as described above. Both the BUI and CLI enable administrators to browse replicated projects, shares, snapshots, and properties much like local projects and shares. However, because replicated shares must exactly match their counterparts on the source ZFSSA, 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 ZFSSAs. 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 ZFSSA 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 ZFSSA from the source.
- Automatic snapshot policies. Automatic snapshot policies can be changed on the target system but these changes have no effect until the package is severed. Automatic snapshots are not taken or destroyed on replicated projects and shares.

The BUI and CLI don't allow administrators to change immutable properties. For shares, a different icon is used to indicate that the property's inheritance cannot be changed:

FIGURE 13-2 Managing Replication Package Properties



Properties ☑ Inherit from project

Export □

Mountpoint /export/afs1

Read only □

Update access time on read ☑

Non-blocking mandatory locking □

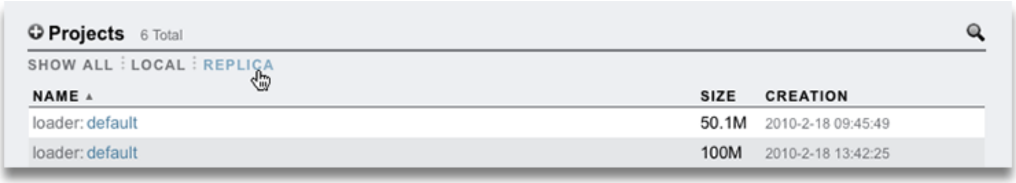
Deferred updates provided with the 2010.Q1 release must be applied on replication targets in order to modify properties on such targets. The system will not allow administrators to modify properties inside replication packages on systems which have not applied the 2010.Q1 deferred updates.

The current release does not support configuration of "chained" replication (that is, replicating replicated shares to another ZFSSA).

Managing Replication Packages in the BUI

Replication packages are displayed in the BUI as projects under the "Replica" filter:

FIGURE 13-3 Replica Filter



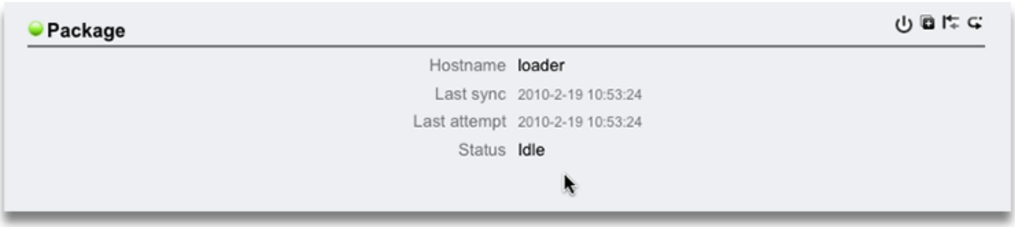
Projects 6 Total

SHOW ALL : LOCAL : REPLICA

NAME ▲	SIZE	CREATION
loader: default	50.1M	2010-2-18 09:45:49
loader: default	100M	2010-2-18 13:42:25

Selecting a replication package for editing brings the administrator to the Shares view for the package's project. From here, administrators can manage replicated shares much like local shares with the exceptions described above. Package properties (including status) can be modified under the Replication tab:

FIGURE 13-4 Shares View for the Package's Project

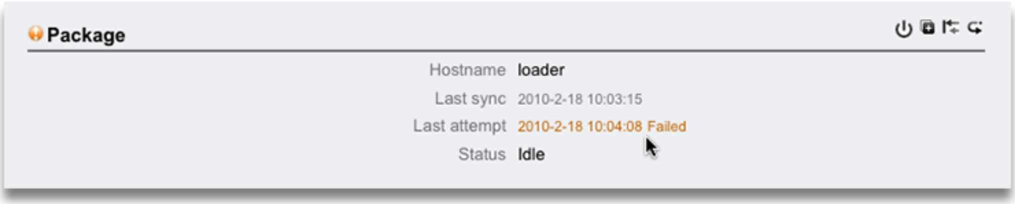


Package

Hostname loader
Last sync 2010-2-19 10:53:24
Last attempt 2010-2-19 10:53:24
Status Idle

The status icon on the left changes when replication has failed:

FIGURE 13-5 Status Icon Indicates Failure



Package

Hostname loader
Last sync 2010-2-18 10:03:15
Last attempt 2010-2-18 10:04:08 Failed
Status Idle

Packages are only displayed in the BUI after the first replication update has begun. They may not appear in the list until some time after the first update has completed.

Managing Replication Packages in the CLI

Replication packages are organized in the CLI by source under `shares replication sources`. Administrators first select a source, then a package. Package-level operations can be performed on this node, or the project can be selected to manage project properties and shares just like local projects and shares with the exceptions described above. For example:

```
loader:> shares replication sources
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
```

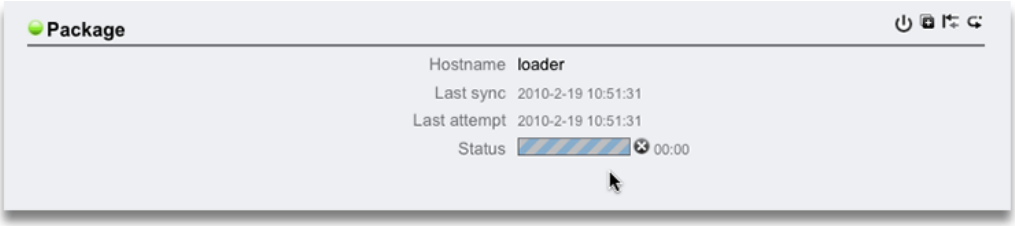
You can display replication sources from `configuration services replication` also. For example:

```
loader:configuration services replication> show
Properties:
                     <status> = online
Children:
                         targets => Configure replication targets
                         sources => View and manage replication packages
```

Canceling Replication Updates

To cancel in-progress replication updates on the target using the BUI, navigate to the replication package (see above), then click the Replication tab. If an update is in progress, you will see a barber pole progress bar with a cancel button (⊗) next to it as shown here:

FIGURE 13-6 Canceling Replication


Package

Hostname loader
Last sync 2010-2-19 10:51:31
Last attempt 2010-2-19 10:51:31
Status 00:00

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 ZFSSA 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" on page 326). 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. 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

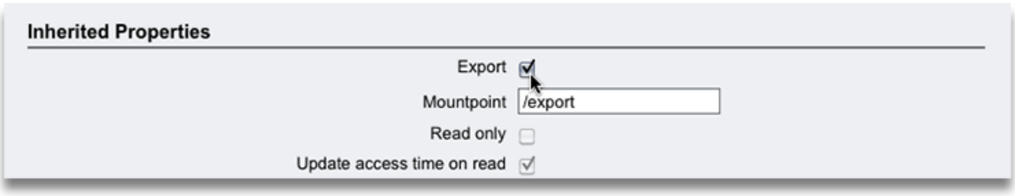
Replicated filesystems can be exported read-only to NAS clients. This can be used to verify the replicated data or to perform backups or other intensive operations on the replicated data (offloading such work from the source ZFSSA).

The filesystem's contents always match the most recently received replication snapshot for that filesystem. This may be newer than the most recently received snapshot for the entire package, and it may not match the most recent snapshot for other shares in the same package. See “ Snapshots and Data Consistency ” on page 381 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:

FIGURE 13-7 Inherited Properties



Inherited Properties

Export ☑

Mountpoint /export

Read only □

Update access time on read ☑

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 ZFSSAs or in other scenarios that don't involve replicating the received data back to the source as part of a typical two-system disaster recovery plan.

Replication can be severed from the BUI by navigating to the replication package (see above), clicking the Replication tab, and clicking the ⇤ button. The resulting dialog allows the administrator to specify the name of the new local project.

Replication can be severed from the CLI by navigating to the replication package (see above), and using the `sever` command. This command takes an optional argument specifying the name of the new local project. If no argument is specified, the original name is used.

Because all local shares are exported, all shares in a package are exported when the package is severed, whether or not they were previously exported (see above). If there are mountpoint conflicts between replicated filesystems and other filesystems on the system, the sever operation will fail. These conflicts must be resolved before severing by reconfiguring the mountpoints of the relevant shares.

Reversing the Direction of Replication

The direction of the replication can be reversed to support typical two-system disaster recovery plans. This operation is similar to the sever operation described above, but additionally configures a replication action on the new local project for incremental replication back to the source system. No changes are made on the source system when this operation is completed, but the first update attempt using this action will convert the original project on the source system into a replication package and rollback any changes made since the last successful replication update from that system.

This feature does not automatically redirect production workloads, failover IP addresses, or perform other activities related to the disaster-recovery failover besides modifying the read-write status of the primary and secondary data copies.

As part of the conversion of the original source project into a replication package on the original source system (now acting as the target), the shares that were replicated as part of the action/package currently being reversed are moved into a new replication package and unexported. The original project remains in the local collection but may end up empty if the action/package included all of its shares. When share-level replication is reversed, any other shares in the original project remain unchanged.

After establishing share-level replication from one ZFSSA to another, reversing that replication on the target ZFSSA destroys the replication schedule. A replication action is then created at the project level which contains the correct target ZFSSA without a schedule.

As mentioned above, this feature is typically used to implement a two-system disaster recovery configuration in which a *primary* system serves production data and replicates it to a *secondary* or *DR* system (often in another data center) standing by to take over the production traffic in the event of a disaster at the primary site. In the event of a disaster at the primary site, the secondary site's copy must be made "primary" by making it writable and redirecting production traffic to the secondary site. When the primary site is repaired, the changes accumulated at the secondary site can be replicated back to the primary site and that site can resume servicing the production workload.

A typical sequence of events under such a plan is as follows:

- The primary system is serving the production workload and replicating to the secondary system.
- A disaster occurs, possibly representing a total system failure at the primary site. Administrators reverse the direction of replication on the secondary site, exporting the replicated shares under a new project configured for replication back to the primary site for when primary service is restored. In the meantime, the production workload is redirected to the secondary site.
- When the primary site is brought back online, an administrator initiates a replication update from the secondary site to the primary site. This converts the primary's copy into a replication package, rolling back any changes made since the last successful update to

the target (before the failure). When the primary site's copy is up-to-date, the administrator reverses the direction of replication again, making the copy at the primary site writable. Production traffic is redirected back to the primary site. Replication is resumed from the primary to the secondary, restoring the initial relationship between the primary and secondary copies.

When reversing the direction of replication for a package, it is strongly recommended that administrators first stop replication of that project from the source. If a replication update is in progress when an administrator reverses the direction of replication for a project, administrators cannot know which consistent replication snapshot was used to create the resulting project on the former target ZFSSA (now source ZFSSA).

Replication can be reversed from the BUI by navigating to the replication package (see above), clicking the Replication tab, and clicking the ↪ button. The resulting dialog allows the administrator to specify the name of the new local project.

Replication can be reversed from the CLI by navigating to the replication package (see above), and using the `reverse` command. This command takes an optional argument specifying the name of the new local project. If no argument is specified, the original name is used.

Because all local shares are exported, all shares in a package are exported when the package is reversed, whether or not they were previously exported (see above). If there are mount point conflicts between replicated filesystems and other filesystems on the system, the reverse operation will fail. These conflicts must be resolved before severing by reconfiguring the mount points of the relevant shares. Because this operation is typically part of the critical path of restoring production service, it is strongly recommended to resolve these mount point conflicts when the systems are first set up rather than at the time of DR failover.

Destroying a Replication Package

The project and shares within a package cannot be destroyed without destroying the entire package. The entire package can be destroyed from the BUI by destroying the corresponding project. A package can be destroyed from the CLI using the destroy command at the `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.

Replication Tasks

The following tasks are examples of replication procedures.

Reversing Replication - Establish Replication

The following is an example of reversing replication to support typical two-system disaster recovery. In this example M11 is the production system and M5 is the recovery system.

▼ Reverse Replication

1. **On Production System M11, navigate to Configuration > SERVICES.**
2. **On the SMB line under Data Services, if the status is Disabled, click Enable service.**
3. **Navigate to Configuration > SERVICES > Remote Replication.**
4. **Click ⊕ Targets and configure name, hostname and password settings. Name=M5, Host name=192.168.1.17, Root password=pppp$1234**
5. **Select Pool=Pool1.**
6. **Navigate to Shares > PROJECTS.**
7. **Click ⊕ Projects. Name=P1**
8. **Navigate to Shares > PROJECTS > P1 > Protocols.**
9. **In the SMB section, set Resource Name=on.**
10. **Navigate to Shares > PROJECTS > P1 > Shares.**
11. **Click ⊕ FileSystems. Name=S1, User=root, Group=other, Permissions=RWX RWX RWX**
12. **Navigate to Shares > PROJECTS > P1 > Shares > S1 > Protocols. The SMB section shows that S1 can be reached using SMB at \\192.168.1.7\S1.**
13. **Navigate to Shares > PROJECTS > P1 > Replication.**
14. **Click ⊕ Actions and set the target and pool. Target=M5, Pool=Pool1**
15. **Click ⊕ Schedule and set the frequency. Frequency=Half-Hour at 00 minutes past the hour.**

16. **On SMB Client System, Map network drive \\192.168.1.7\S1 (user=root, password=pppp$1234).**

17. **Create file F1.txt.**

18. **On Production System M11, navigate to Shares > PROJECTS > P1 > Replication.**

19. **On the Action line TARGET=M5, click Update now.**

20. **After replication is finished, click Disable, STATUS changes to disabled.**

Reversing Replication - Simulate Recovery from a Disaster

To simulate recovery from a disaster that prevents access to the production system, use the recovery system to reverse replication. When replication is reversed, the replication package present on the target is converted to a local project and additionally a replication action is configured for this local project for incremental replication back to the original source system. This replication action is not enabled by default. The administrator should send the update manually.

▼ Reverse Replication

1. **To simulate loss of contact with Primary System M11 on the SMB Client System, select Disconnect network drive.**

2. **On Disaster Recovery System M5 select Pool=Pool1.**

3. **Navigate to Shares > PROJECTS > REPLICA. The Project M11:P1 is listed**

4. **Navigate to Shares > PROJECTS > REPLICA> M11:P1 > Replication. The package has the Status=Idle**

5. **Click Reverse the direction of replication and set the new project name. New Project Name=P1**

6. **Navigate to Shares > PROJECTS > REPLICA. Project M11:P1 is not listed anymore because the replication package present on the target is converted to a local project.**

7. **Navigate to Configuration > SERVICES.**

8. **On the SMB line under Data Services, if the status is Disabled, click Enable service.**

9. **Navigate to Shares > PROJECTS > LOCAL. Project P1 is listed.**

10. **Navigate to Shares > PROJECTS > P1 > Protocols.**

11. **In the SMB section, set Resource Name=on.**

12. **Navigate to Shares > PROJECTS > P1 > Shares > S1 > Protocols. The SMB section shows that S1 can be reached using SMB at \\192.168.1.17\S1.**

13. **On SMB Client System, Map network drive \\192.168.1.17\S1 (user=root, password=pppp$1234).**

14. **Edit file F1.txt and save it as F2.txt. NOTE: During a real disaster recovery sequence, once communications with Production System M11 were restored, you would have the opportunity to trigger manual, scheduled, or continuous replication updates while applications continue to access data on Disaster Recovery System M5.**

15. **To prepare for the transition back to the production system, select Disconnect network drive.**

16. **On Disaster Recovery System M5, navigate to Shares > PROJECTS > P1 > Replication.**

17. **On Action line TARGET=M11 click Update now.**

18. **After replication is finished, click Disable.**

Reversing Replication - Resume Replication from Production System

Each time replication is reversed, you provide a new project name that is used when the replication package is converted into a new local project. If you want to use the same name and a prior replication reversal has left behind an empty local project with that name, you must delete the existing empty project so that the next reversal can create a project with the same name.

▼ Reverse Replication

1. **On Production System M11, navigate to Shares > PROJECTS > LOCAL > P1.**
   P1 is empty because the first update from the new source system (the original target) converts the original project on the original source system into a replication package. This example employs a project-level replication action that replicates all of the shares in the project. Therefore, all of the shares that were present under this local project are now present under the replication package, leaving the local project empty.

2. **Navigate to Shares > PROJECTS > LOCAL.**

3. **Delete P1. It is safe to delete this empty project because its contents have been moved to the replication package as a result of the replication reversal from the original target.**

4. **Navigate to Shares > PROJECTS > REPLICA > M5:P1 > Replication.**

5. **Click Reverse the direction of replication and set the project name. New Project Name=P1**

6. **On SMB Client System, Map network drive \\192.168.1.7\S1 (user=root, password=pppp$1234). Files F1.txt and F2.txt are listed.**

7. **On Production System M11, navigate to Shares > PROJECTS > P1 > Replication.**

8. **On Action line TARGET=M5, click Edit entry.**

9. **Click ⊕ Schedule and set the frequency. Frequency=Half-Hour at 00 minutes past the hour.**

10. **On Action line TARGET=M5, click Update now. This directs Disaster Recovery System M5 to convert the its local project P1 back into replication package M11:P1.**

11. **Monitor the UPDATES column on action line TARGET=M5 and wait for the replication update to complete.**

12. **On Disaster Recovery System M5, navigate to Shares > PROJECTS > LOCAL > P1. P1 is empty because the project was converted back into a replication package.**

13. **Navigate to Shares > PROJECTS > LOCAL.**

14. **To enable the next reversal to convert the replication package to a project named P1, delete P1.**

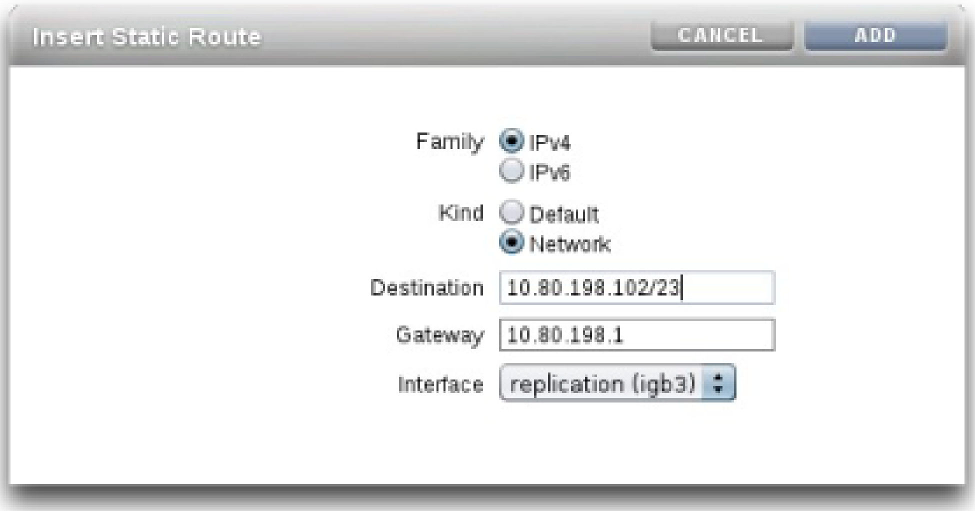
Forcing Replication to use a Static Route

To consolidate replication traffic onto a specific network interface, you must connect the source and target ZFS ZFSSAs using static routes. To set up the static routes, use the following steps:

▼ Force Replication to use a Static Route

1. **To set up a static route, on the Configuration > Network > Routing page, click the ⊕ add icon.**

2. **In the Insert Static Route box, select the Family and Kind and then enter the Destination IP, Gateway, and Interface.**



Family ● IPv4 ○ IPv6

Kind ○ Default ● Network

Destination 10.80.198.102/23

Gateway 10.80.198.1

Interface replication (igb3)

3. **Click Add.**

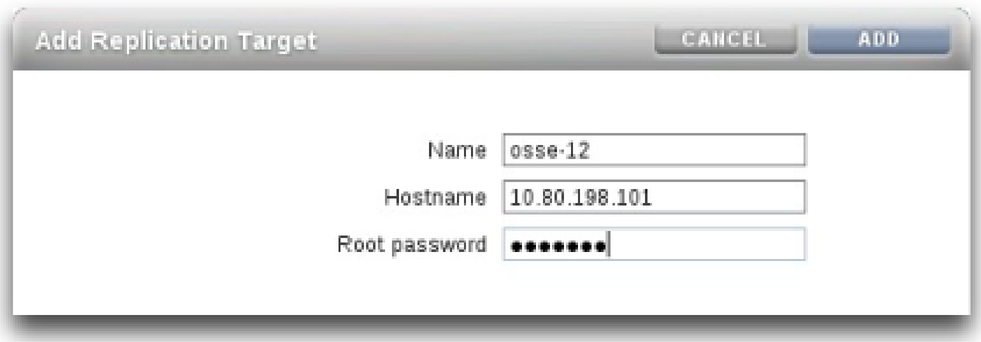
4. **To ensure traffic is routing on the source and target, in the CLI use `traceroute`. For information about using `traceroute`, see “Network Routing**

Configuration" on page 78. In the example, `10.80.219.124 @ igb0` identifies igb0 as the interface. This is a quick way to verify the right interface is being used.

```
brmv01sn02:> traceroute poc7330-050
traceroute: Warning: Multiple interfaces found; using 10.80.219.124 @ igb0 traceroute
 to poc7330-050 (10.80.219.117), 30 hops max, 40 byte packets
 1  poc7330-050.us.oracle.com (10.80.219.117)  0.446 ms  0.115 ms  0.104 ms
```

5. **To add a new replication target, on the Configuration > Services > Replication page, click the ⊕ add icon.**

6. **In the Add Replication Target box, type a name for the target, the Hostname IP address for the network interface, and the password.**



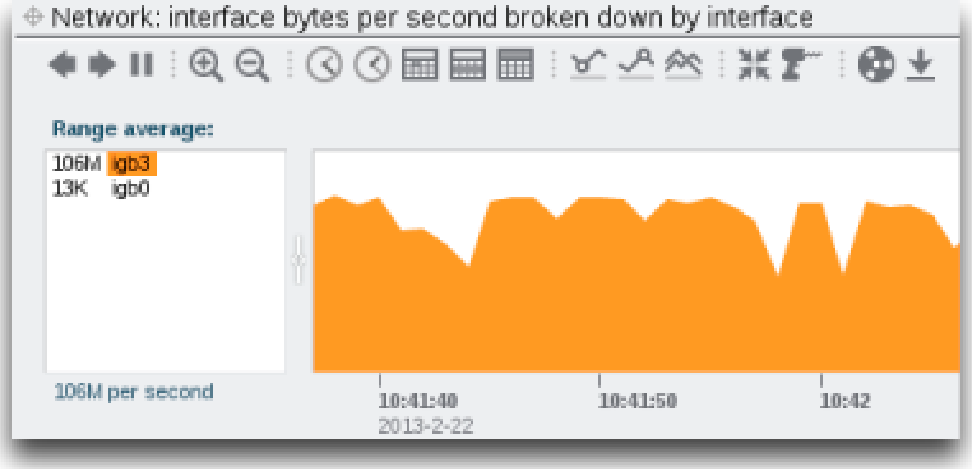
Name: osse-12

Hostname: 10.80.198.101

Root password: ●●●●●●●

7. **Click Add.**

8. **To ensure traffic passes across the defined static route, after replication starts use "Network Interface Bytes" in "Oracle ZFS Storage Appliance Analytics Guide ".**



9. **Preferences page ensure “Preference Properties” on page 141 is toggled on.**

10. **After you verify that source to target replication uses the correct interface, reverse the replication. For information about reversing replication, see “ Reversing the Direction of Replication ” on page 370.**

Cloning a Received Replication Project

This is a CLI 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> 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 “User Authorizations” on page 132 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.
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

Authorization	Details
	authorizations in the Projects and Shares scope for details.

The rrsource authorization is required to configure replication targets on an ZFSSA, even though this is configured under the Remote Replication service screen. For help with authorizations, see “User Authorizations” on page 132.

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).
- When a Continuous replication starts for the first time.
- When a Continuous replication fails.
- When a continuous replication starts for the first time, fails, or resumes after a failure.

Replication Audit Events

The system audits the following replication events and records them in the “Logs” in “Oracle ZFS Storage Appliance Customer Service Manual ”.

- Creating, modifying or destroying replication actions
- Adding or removing shares from a replication group
- Creating, modifying, cloning, reversing, severing or destroying replication packages on the target
- Creating, modifying or destroying replication targets

Replication and Clustering

Replication can be configured from any ZFS Storage Appliance to any other ZFS Storage Appliance regardless of whether each is part of a cluster and whether the ZFSSA's cluster peer has replication configured in either direction, except for the following constraints:

- Configuring replication from both peers of a cluster to the same replication target is unsupported, but a similar configuration can be achieved using two different IP addresses for the same target ZFSSA. Administrators can use the multiple IP addresses of the target ZFSSA to create one replication target on each cluster head for use by that head.

- When configuring replication between cluster peers, configure replication with both controllers in the CLUSTERED state. Do not use private network addresses and use separate replication targets for each controller's pools.

The following rules govern the behavior of replication in clustered configurations:

- Replication updates for projects and shares are sent from whichever cluster peer has imported the containing storage pool.
- Replication updates are received by whichever peer has imported the IP address configured in the replication action on the source. Administrators must ensure that the head using this IP address will always have the storage pool containing the replica imported. This is ensured by assigning the pool and IP address resources to the same head during cluster configuration.
- Replication updates (both to and from an ZFSSA) that are in progress when an ZFSSA 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 Chapter 10, “Cluster Configuration”.

Snapshots and Data Consistency

The ZFSSA replicates snapshots and each snapshot is received atomically on the target, so the contents of a share's replica on the target always matches the share's contents on the source at the time the snapshot was taken. Because the snapshots for all shares sent in a particular group are taken at the same time (see above), the entire package contents after the completion of a successful replication update exactly matches the group's content when the snapshot was created on the source (when the replication update began).

However, each share's snapshots are replicated separately (and serially), so it's possible for some shares within a package to have been updated with a snapshot that is more recent than those of other shares in the same package. This is true during a replication update (after some shares have been updated but before others have) and after a failed replication update (after which some shares may have been updated but others may not have been).

To summarize:

- Each share is always point-in-time consistent on the target (self-consistent).

- When no replication update is in progress and the previous replication update succeeded, each package's shares are also point-in-time consistent with each other (package-consistent).
- When a replication update is in progress or the previous update failed, package shares may be inconsistent with each other, but each one will still be self-consistent. If package consistency is important for an application, one must clone the replication package, which always clones the most recent successfully received snapshot of each share.

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:

- Determine whether this is an incremental or full update based on whether we've tried to replicate this action before and whether the target already has the necessary snapshot for an incremental update.
- Take a new project-level snapshot.
- Send the update. For a full update, send the entire group's contents up to the new snapshot. For an incremental update, send the difference between from the previous (base) snapshot and the new snapshot.
- Record the new snapshot as the base snapshot for the next update and destroy the previous base snapshot (for incremental updates). The base snapshot remains on the target until the next update is received at which point it is the first thing that is destroyed.

This has several consequences for snapshot management:

- During the first replication update and after the initial update when replication is not active, there is exactly one project-level snapshot for each action configured on the project or any share in the group. A replication action may create snapshots on shares that are in the same project as the share(s) in the group being replicated by the action but that are not being sent as part of the update for the group.
- During subsequent replication updates of a particular action, there may be two project-level snapshots associated with the action. Both snapshots may remain after the update completes in the event of failure where the source was unable to determine whether the target successfully received the new snapshot (as in the case of a network outage during the update that causes a failure).
- None of the snapshots associated with a replication action can be destroyed by the administrator without breaking incremental replication. The system will not allow

administrators to destroy snapshots on either the source or target that are necessary for incremental replication. To destroy such snapshots on the source, one must destroy the action (which destroys the snapshots associated with the action). To destroy such snapshots on the target, one must first sever the package (which destroys the ability to receive incremental updates to that package).

- Administrators must not rollback to snapshots created prior to any replication snapshots. Doing so will destroy the later replication snapshots and break incremental replication for any actions using those snapshots.
- Replication's usage of snapshots requires that administrators using replication understand "space management" on page 283 on the ZFSSA, particularly "as it applies to snapshots" on page 283.
- For information about space management for replicating LUNs, see "Space Management for Replicating LUNs" on page 283

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 ZFSSA will have the same SCSI GUID as the LUN on the source ZFSSA. 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 ZFSSA management software, but attempting to replicate a clone when the origin snapshot does not exist on the target will fail.

In practice, there are several ways to ensure that replication of a clone will succeed:

- If the clone's origin snapshot is in the same project, just use project-level replication.

- If the clone's origin snapshot is not in the same project or if project-level replication that includes the origin is undesirable for other reasons, use share-level replication to replicate the origin share first and then use project-level or share-level replication to replicate the clone.
- Do not destroy the clone's origin on the target system unless you intend to also destroy the clone itself.

In all cases, the "include snapshots" property should be true on the origin's action to ensure that the origin snapshot is actually sent to the target.

Observing Replication

The following “Analytics” in “Oracle ZFS Storage Appliance Analytics Guide ” are available for replication:

- “Data Movement Replication Operations” in “Oracle ZFS Storage Appliance Analytics Guide ”
- “Data Movement Replication Bytes” in “Oracle ZFS Storage Appliance Analytics Guide ”
- “Statistics” in “Oracle ZFS Storage Appliance Analytics Guide ” are also available.

Replication Failures

Individual replication updates can fail for a number of reasons. Where possible, the ZFSSA reports the reason for the failure in alerts posted on the source ZFSSA or target ZFSSA, 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 ZFSSA was unable to connect to the target ZFSSA due to a network problem. There may be a misconfiguration on the source, target, or the network.
Peer verification failed	The ZFSSA 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 ZFSSA for a target which has been reinstalled or factory reset in order to generate a new set of authentication keys. See “Project Replication Targets ” on page 353.

Failure	Details
Peer RPC failed	A remote procedure call failed on the target system. This occurs most commonly when the target ZFSSA is running incompatible software. For details, see “ Upgrading From 2009.Q3 and Earlier ” on page 387.
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 ZFSSA.
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 ZFSSA 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.
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.

Failure	Details
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.

Replication Compatibility

Before performing a replication update, the replication service verifies that the target system is compatible with new data from the source.

- If there are features in use on the source that are not compatible with the target and the features can be safely disabled, the replication service will disable the features, perform the update, and issue a warning.
- If there are features in use on the source that are not compatible with the target and cannot be disabled, the replication service will not perform the update and issue an error.

NOTE: It is always best to upgrade the target as soon as possible.

Updates that break replication compatibility, are delivered as Deferred Updates. For the current list and description, see “Updates” in “Oracle ZFS Storage Appliance Customer Service Manual ” and the Oracle ZFS Storage Appliance Release Notes for your current release.

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 ZFSSA 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.

♦ ♦ ♦ CHAPTER 14

Shadow Migration

This section describes shadow migration for the ZFSSA.

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 ZFSSA has two integrated solutions for migrating data that addresses the most common use cases. The first, Chapter 13, "Replication", is intended for replicating data between one or more ZFSSAs, 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 new ZFSSA 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.

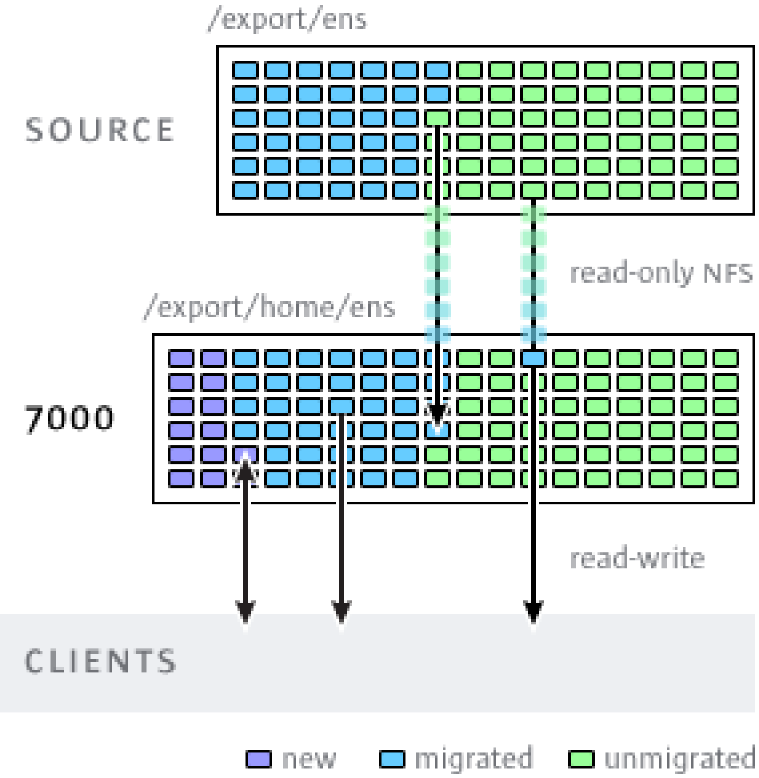
Migration via External Interposition

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

Shadow Migration

FIGURE 14-1 Shadow Migration



Shadow migration uses interposition, but is integrated into the ZFSSA 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 ZFSSA 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 ZFSSA. Clients can then access the ZFSSA in read-write mode.

Once the shadow property is set, data is transparently migrated in the background from the source ZFSSA locally. If a request comes from a client for a file that has not yet been migrated, the ZFSSA 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 ZFSSA 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 ZFSSA.
- 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 File System 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 ZFSSA 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 ZFSSA must have the same name service configuration.
- The migration source and target ZFSSA 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 ZFSSA.

If you see files or directories owned by "nobody", it likely means that the ZFSSA 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" on page 229. There is a single global tunable which is the number of threads dedicated to this task. Increasing the number of threads will result in greater parallelism at the expense of additional resources.

The shadow migration service can be disabled, but this should only be used for testing purposes, or when the active of shadow migration is overwhelming the system to the point where it needs to be temporarily stopped. When the shadow migration service is disabled, synchronous requests are still migrated as needed, but no background migration occurs. With the service disabled no shadow migration will ever complete, even if all the contents of the filesystem are read manually. It is highly recommended to always leave the service enabled.

Handling Migration Errors

Because shadow migration requires committing new writes to the server prior to migration being complete, it is very important to test migration and monitor for any errors. Errors encountered during background migration are kept and displayed in the BUI as part of shadow migration status. Errors encountered during other synchronous migration are not tracked, but will be accounted for once the background process accesses the affected file. For each file, the remote filename as well as the specific error are kept. Clicking on the information icon next to the error count will bring up this detailed list. The error list is not updated as errors are fixed, but simply cleared by virtue of the migration completing successfully.

Shadow migration will not complete until all files are migrated successfully. If there are errors, the background migration will continually retry the migration until it succeeds. This allows the administrator to fix any errors (such as permission problems), let the migration complete, and be assured of success. If the migration cannot complete due to persistent errors, the migration can be canceled, leaving the local filesystem with whatever data was able to be migrated. This should only be used as a last resort - once migration has been canceled, it cannot be resumed.

Monitoring Migration Progress

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 ZFSSA 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 ZFSSA 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 File Systems

Shadow filesystems can be snapshotted, however the state of what is included in the snapshot is arbitrary. Files that have not yet been migrated will not be present, and implementation details (such as SUNWshadow extended attributes) may be visible in the snapshot. This snapshot can be used to restore individual files that have been migrated or modified since the original migration began. Because of this, it is recommended that any snapshots be kept on the source until the migration is completed, so that unmigrated files can still be retrieved from the source if necessary. Depending on the retention policy, it may be necessary to extend retention on the source in order to meet service requirements.

While snapshots can be taken, these snapshots cannot be rolled back to, nor can they be the source of a clone. This reflects the inconsistent state of the on-disk data during the migration.

Backing Up Shadow File Systems

Filesystems that are actively migrating shadow data can be backed using NDMP as with any other filesystem. The shadow setting is preserved with the backup stream, but will be restored only if a complete restore of the filesystem is done and the share doesn't already exist. Restoring individual files from such a backup stream or restoring into existing filesystems may result in inconsistent state or data corruption. During the full filesystem restore, the filesystem will be in an inconsistent state (beyond the normal inconsistency of a partial restore) and shadow migration will not be active. Only when the restore is completed is the shadow setting restored. If the shadow source is no longer present or has moved, the administrator can observe any errors and correct them as necessary.

Replicating Shadow File Systems

Filesystems that are actively migrating shadow data can be replicated using the normal mechanism, but only the migrated data is sent in the data stream. As such, the remote side contains only partial data that may represent an inconsistent state. The shadow setting is sent along with the replication stream, so when the remote target is failed over, it will keep the same shadow setting. As with restoring an NDMP backup stream, this setting may be incorrect in the context of the remote target. After failing over the target, the administrator can observe any errors and correct the shadow setting as necessary for the new environment.

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" in "Oracle ZFS Storage Appliance Analytics Guide ". 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.

Migrating Local File Systems

In addition to its primary purpose of migrating data from remote sources, the same mechanism can also be used to migrate data from local filesystem to another on the ZFSSA. 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.

Shadow Migration Tasks

Before attempting a complete migration, it is important to test the migration to make sure that the ZFSSA has appropriate permissions and security attributes are translated correctly. Once you are confident that the basic setup is functional, the shares can be setup for the final migration.

▼ Testing Potential Shadow Migration

1. **Configure the source so that the ZFSSA 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

1. **Schedule downtime during which clients can be quiesced and reconfigured to point to a new server.**

2. **Configure the source so that the ZFSSA 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.

♦ ♦ ♦ C H A P T E R 15

CLI Scripting

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

Automating Access

You can use “Batching Commands” on page 401 or “Scripting Commands” on page 402 (or some combination), but in any case the automated infrastructure requires automated access to the appliance. This must be done by Chapter 7, “User Configuration”, “User Authorizations” on page 132, and “Setting SSH Public Keys Using the CLI” on page 142.

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 Commands

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

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

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:

TABLE 15-1 Built-in Functions to Support System Interactions

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.
`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.
`choicies`	Returns an array of the valid property values for any property for which the set of values is known and enumerable.

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]);
                try {
                        printf("%-10s %s\n", svcs[i], get('<status>'));
                } catch (err) { }
                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
```

The Choices Function

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

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

Here is the output of running the script:

```
SHARE                                    USED            AVAILABLE
pond:projectA/fs1                        31744           566196178944
pond:projectA/fs2                        31744           566196178944
pond:projectB/lun1                       21474836480     587670999040
puddle:deptA/share1                      238475          467539219283
puddle:deptB/share1                      129564          467539219283
puddle:deptB/share2                      19283747        467539219283
```

Generating Output

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

TABLE 15-2 Built-in Functions for Generating 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.

Function	Description
`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
```

♦ ♦ ♦ CHAPTER 16

Maintenance Workflows

A workflow is a Chapter 15, “CLI Scripting” that is uploaded to and managed by the ZFSSA 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 Chapter 9, “Alert Configuration” or at a designated time. As such, workflows allow for the ZFSSA to be *extended* in ways that capture specific policies and procedures, and can be used (for example) to formally encode best practices for a particular organization or application.

Using Workflows

A workflow is embodied in a valid ECMAscript file, containing a single global variable, `workflow`. This is an Object that must contain at least three members:

TABLE 16-1 Required Object 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 ZFSSA shell, running (by default) as the user executing the workflow. As such, workflows have at their disposal the Chapter 15, “CLI Scripting”, and may interact with the ZFSSA just as any other instance of the ZFSSA 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:

TABLE 16-2 Required Workflow Parameter 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:

TABLE 16-3 Member Type Names

Type name	Description
Boolean	A boolean value
ChooseOne	One of a number of specified values
EmailAddress	An e-mail address

Type name	Description
`File`	A file to be transferred to the ZFSSA
`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:

TABLE 16-4 Constrained Parameters Required Members

Required Member	Type	Description
`options`	Array	An array of strings that specifies the valid options
`optionlabels`	Array	An array of strings that specifies the labels associated with the options specified in `options`

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:

TABLE 16-5 Required Members for Optional Parameters

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`.

Workflow Error Handling

If, in the course of executing a workflow, an error is encountered, an exception will be thrown. If the exception is not caught by the workflow itself (or if the workflow throws an exception that is not otherwise caught), the workflow will fail, and the information regarding the exception will be displayed to the user. To properly handle errors, exceptions should be caught and processed. For example, in the previous example, an attempt to create a share in a non-existent project results in an uncaught exception. 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 + '"');
 }
};
```

Workflow Input validation

Workflows may optionally validate their input by adding a `validate` member that takes as a parameter an object that contains the workflow parameters as members. The `validate` function should return an object where each member is named with the parameter that failed validation, and each member's value is the validation failure message to be displayed to the user. 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 + '"');
 }
};
```

Workflow 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.

Workflow 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:

TABLE 16-6 Required Members for Execution Reporting

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 ZFSSA software that the workflow depends on, and the second is the expression of the version of the workflow itself. Versioning is expressed through two optional members to the workflow:

TABLE 16-7 Optional Members for Versioning

Optional Member	Type	Description
`required`	String	The minimum version of the ZFSSA 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 ZFSSA software, add the optional `required` field to your workflow. The ZFSSA 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 ZFSSA kit software, run the "`configuration version get version`" CLI command, or look at the "Version" field in the "System" in "Oracle ZFS Storage Appliance Customer Service Manual " 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 ZFSSA, 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 ZFSSA 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 Chapter 9, "Alert Configuration". 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:

TABLE 16-8 Required Members for Alert Execution Context

Required Member	Type	Description
`class`	String	The class of the alert.
`code`	String	The code of the alert.

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

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

TABLE 16-9 Required Members for the Items Member

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 Slert 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');
       }
};
```

Using Scheduled Workflows

Workflows can be started via a timer event by setting up a schedule for them. The property scheduled has to be added to the Workflow Object and needs to be set to true. Schedules can either be created via the CLI once a workflow is loaded into the ZFSSA or an array type property named schedule can be added to the Object Workflow.

Using the CLI

Once a workflow has been loaded into the ZFSSA a schedule can be defined for it via the CLI interface as follows:

```
dory:> maintenance workflows
dory:maintenance workflows> "select workflow-002'''
dory:maintenance workflow-002> schedules
dory:maintenance workflow-002 schedules>create
dory:maintenance workflow-002 schedule (uncommitted)> set frequency=day
                    frequency = day (uncommitted)
dory:maintenance workflow-002 schedule (uncommitted)> set hour=10
```

```
                         hour = 10 (uncommitted)
dory:maintenance workflow-002 schedule (uncommitted)> set minute=05
                       minute = 05 (uncommitted)
dory:maintenance workflow-002 schedule (uncommitted)> commit
dory:maintenance workflow-002 schedules> list
NAME                 FREQUENCY            DAY                  HH:MM
schedule-001         day                  -                    10:05
dory:maintenance workflow-002 schedules> create
dory:maintenance workflow-002 schedule (uncommitted)> set frequency=week
                    frequency = week (uncommitted)
dory:maintenance workflow-002 schedule (uncommitted)> set day=Monday
                           day = Monday (uncommitted)
dory:maintenance workflow-002 schedule (uncommitted)> set hour=13
                          hour = 13 (uncommitted)
dory:maintenance workflow-002 schedule (uncommitted)> set minute=15
                       minute = 15 (uncommitted)
dory:maintenance workflow-002 schedule (uncommitted)> commit
dory:maintenance workflow-002 schedules> list
NAME                 FREQUENCY            DAY                  HH:MM
schedule-001         day                             -                      10:05
schedule-002         week                     Monday               13:15
dory:maintenance workflow-002 schedules>
```

Each schedule entry consists of the following properties:

TABLE 16-10 Schedule Properties

Property	Type	Description
NAME	String	Name of the schedule, system generated
frequency	String	minute,halfhour,hour,day,week, month
day	String	Specifies specific day and can be set to: Monday, Tuesday,Wednesday, Thursday,Friday,Saturday or Sunday. Can be set when frequency is set to week or month
hour	String	00-23, Specifies the hour part of the schedule and can be specified when the frequency is set to a day,week or month.
minute	String	00-59, Specifies the minute part of the schedule.

Coding the Schedule

Schedules can also be specified in the workflow code as a property in the Object workflow. The property syntax used here differs from the CLI schedule creation. Here three properties are used,

TABLE 16-11 Schedule Properties

Property	Type	Description
`offset`	Number	Determines the starting point in the defined period
`period`	Number	Defines the frequency of the Schedule
`unit`	String	Specifies if either seconds or month are used as unit in the offset and period definition

The following code example illustrates the use of the properties. Note that inline arithmetic helps to make the offset and period declarations more readable.

```
// Example of using Schedule definitions within a workflow
var MyTextObject = {
 MyVersion: '1.0',
 MyName:  'Example 9',
 MyDescription:  'Example of use of Timer',
 Origin:  'Oracle'
 };
var MySchedules = [
 // half hr interval
 { offset: 0, period: 1800, units: "seconds" },
 // offset 2 days, 4hr, 30min , week interval
 {offset: 2*24*60*60+4*60*60+30*60, period: 604800,units: "seconds" }
];
var workflow = {
 name:  MyTextObject.MyName,
 description: MyTextObject.MyDescription,
 version: MyTextObject.MyVersion,
 alert:  false,
 setid:  true,
 schedules:  MySchedules,
 scheduled: true,
 origin:  MyTextObject.Origin,
 execute: function () {
    audit('workflow started for timer; ');
     }
   }
 };
```

The property units in the Object MySchedules specifies the type of units used for the properties offset and period. They can be set to either seconds or month. The property period specifies the frequency of the event and the offset specifies the units within the period. In the above example the period in the second schedule is set for a week, starting at the second day, at 4:30. Multiple schedules can be defined in the property schedules.

The Object MySchedules in the example uses the following three properties:

- offset - This is the starting offset from January 1, 1970 for the schedule. The offset is given in the units defined by the property "units".
- period - This is the period between recurrences of the schedule which is also given in the units defined by the property "units."
- units - This can be defined in seconds or months.

The starting point for weekly schedules is Thursday. This is due to the fact that the epoch is defined as starting on 1 Jan 1970 which was a Thursday.

In the above example the period in the second schedule uses a starting offset of 2 days + 4 hours + 30 minutes. This results in the starting date being January 3, 1970 at 4:30 am. The schedule recurs weekly indefinitely every Saturday at 4:30 am. Below you can see the display of the schedule in the CLI.

```
<small>dory:> maintenance workflows
dory:maintenance workflows> list
WORKFLOW     NAME                       OWNER SETID ORIGIN               VERSION
workflow-000 Configure for Oracle Solaris Cluster NFS root  false Oracle Corporation   1.0.0
workflow-001 Unconfigure Oracle Solaris Cluster NFS root  false Oracle Corporation   1.0.0
workflow-002 Configure for Oracle Enterprise Manager Monitoring root  false Sun Microsystems,
 Inc. 1.1
workflow-003 Unconfigure Oracle Enterprise Manager Monitoring root  false Sun Microsystems,
 Inc. 1.0</small>
```

dory:maintenance workflow-002 schedules>

```
NAME                FREQUENCY           DAY                 HH:MM
schedule-000        halfhour            -                   --:00
schedule-001        week                Saturday            04:30
```

Example: device type selection

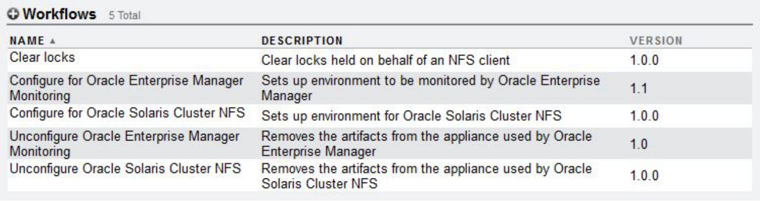
Here is an example workflow that creates a worksheet based on a specified drive type:

```
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 ZFSSA by clicking on the plus icon, and they are executed by clicking on the row specifying the workflow.

FIGURE 16-1



⊕ Workflows 5 Total

NAME ▲	DESCRIPTION	VERSION
Clear locks	Clear locks held on behalf of an NFS client	1.0.0
Configure for Oracle Enterprise Manager Monitoring	Sets up environment to be monitored by Oracle Enterprise Manager	1.1
Configure for Oracle Solaris Cluster NFS	Sets up environment for Oracle Solaris Cluster NFS	1.0.0
Unconfigure Oracle Enterprise Manager Monitoring	Removes the artifacts from the appliance used by Oracle Enterprise Manager	1.0
Unconfigure Oracle Solaris Cluster NFS	Removes the artifacts from the appliance used by Oracle Solaris Cluster NFS	1.0.0

CLI

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

Downloading workflows

Workflows are downloaded to the ZFSSA via the `download` command, which is similar to the “System” in “Oracle ZFS Storage Appliance Customer Service Manual ”:

```
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:

```
<small>dory:maintenance workflows> list
WORKFLOW     NAME                       OWNER SETID ORIGIN               VERSION
workflow-000 Configure for Oracle Solaris Cluster NFS root  false Oracle Corporation   1.0.0
workflow-001 Unconfigure Oracle Solaris Cluster NFS root  false Oracle Corporation   1.0.0
workflow-002 Configure for Oracle Enterprise Manager Monitoring root  false Sun Microsystems,
 Inc. 1.1
workflow-003 Unconfigure Oracle Enterprise Manager Monitoring root  false Sun Microsystems,
 Inc. 1.0</small>
```

To view workflows, use the `show` command from the `maintenance workflows` context:

```
dory:maintenance workflows> select workflow-001
dory:maintenance workflow-001> show
Properties:
                          name = Configure for Oracle Solaris Cluster NFS
                   description = Sets up environment for Oracle Solaris Cluster NFS
                         owner = root
                        origin = Oracle Corporation
                         setid = false
                         alert = false
                       version = 1.0.0
                     scheduled = false
```

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
```

♦ ♦ ♦ CHAPTER 17

Integration

Oracle ZFS Storage Appliances deliver a full suite of data protocols to communicate with a wide variety of application hosts. To improve application performance or more tightly integrate with your application environment, follow the best practices outlined in White Papers and Solutions Briefs found on the NAS Storage Documentation page.

- "Symantec DMP/Storage Foundation"

For some applications, installing software on the application host enhances interoperability. The following articles provide an overview of how software integration can provide a better experience for storage administrators. Comprehensive documentation is packaged with each download.

- "Sun ZFS Storage Appliance Plug-in for Oracle Solaris Cluster Geographic Edition" on page 464
- "Sun ZFS Storage Appliance Network File System Plug In for Oracle Solaris Cluster" on page 464
- "Sun ZFS Storage Appliance Provider For Volume Shadow Copy Service Software" on page 466
- "Sun ZFS Storage Management Plug-In for Oracle Enterprise Manager Grid Controller" on page 464
- "Oracle Virtual Machine Storage Connect Plug-in for the Sun ZFS Storage Appliance" on page 466
- "Sun ZFS Storage 7000 Storage Replication Adapter for VMware Site Recovery Manager" on page 468
- " Oracle Intelligent Storage Protocol " on page 463

Your appliance is also uniquely featured to seamlessly integrate with other Oracle products. For instance, the following sections describe how to configure the ZFS Storage Appliance as a backup target for the Oracle Exadata Database Machine and the Oracle SPARC SuperCluster.

- "Oracle Exadata Database Machine Backup" on page 432
- "Configuring Oracle SPARC SuperCluster for ZFS Storage Appliance Backup" on page 459

For more information, visit the NAS Storage Documentation page.

Oracle Exadata Database Machine Backup

When equipped with native QDR InfiniBand and 10Gb Ethernet connectivity options, the ZFS Storage Appliance is ideal for reliably backing up Oracle Exadata. The Oracle Exadata Backup Configuration Utility is provided for deployment using a command-line tool, or your appliance can be configured manually using the instructions in the following sections:

- "Manual Configuration of a Sun ZFS Storage Appliance" on page 432
- "Configuring Oracle Exadata for a Sun ZFS Storage Appliance" on page 436

Comprehensive documentation is packaged with the utility, including instructions for how to execute a backup from the Oracle Exadata. Whether manually or using the utility, configuration of networking and storage pools on the appliance is required in addition to either approach.

For detailed information on using your ZFS Storage Appliance as a backup target for Oracle Exadata, see the Protecting Oracle Exadata with the Sun ZFS Storage Appliance: Configuration Best Practices white paper on the NAS Storage Documentation page. Also available is an Oracle ZFS Storage ZS3-4 cluster, offered pre-racked with disk shelves as the Oracle ZFS Storage ZS3-BA to minimize set-up complexity. Integrating this appliance with Oracle Exadata is identical to the process described above.

Manual Configuration of a Sun ZFS Storage Appliance

This section provides general guidelines for manually configuring a ZFS Storage Appliance for use with the Oracle Exadata. For detailed information, see the Protecting Oracle Exadata with the Sun ZFS Storage Appliance: Configuration Best Practices white paper on the NAS Storage Documentation page.

Configuring Networks, Pools, and Shares

The following sections summarize best practices for optimizing ZFS Storage Appliance network, storage pool, and share configurations to support backup and restore processing.

Network Configuration

This section describes how to configure the IP network multipathing (IPMP) groups, and how to configure routing in the ZFS Storage Appliance.

Note: If you used the Oracle Exadata Backup Configuration Utility, configure the network as described in this section. For details, review the Best Practices white paper.

For customers seeking additional IB connectivity, more IB HCAs can be installed and configured. For details, see the Oracle ZFS Storage Appliance Installation Guide.

The principles in this section can be applied to a 10Gb Ethernet implementation by applying the network configuration to the ixgbe interfaces instead of the ibp interfaces. The 10Gb Ethernet implementation may be configured as active/active IPMP. If the ZFS Storage Appliance is on a different subnet than the Oracle Exadata, it may be necessary to create static routes from the ZFS Storage Appliance to the Oracle Exadata. Consult with your network administrator for details.

▼ Basic Network Configuration

1. **Ensure that the ZFS Storage Appliance is connected to the Oracle Exadata.**
2. **Configure `ibp0`, `ibp1`, `ibp2`, and `ibp3` with address `0.0.0.0/8` (necessary for IPMP), connected mode, and partition key `ffff`. To identify the partition key used by the Oracle Exadata system, run the following command as the root user:<br/>`# cat /sys/class/net/ib0/pkey`**
3. **Configure the active/standby IPMP group over `ibd0` and `ibd3`, with `ibd0` active and `ibd3` standby.**
4. **Configure the active/standby IPMP group over `ibd1` and `ibd2`, with `ibd2` active and `ibd1` standby.**
5. **Enable adaptive routing to ensure traffic is load balanced appropriately when multiple IP addresses on the same subnet are owned by the same head. This occurs after a cluster failover.**

Pool Configuration

This section describes design considerations to determine the most appropriate pool configuration for the ZFS Storage Appliance for Oracle Recovery Manager (RMAN) backup and restore operations based on data protection and performance requirements.

Note: If you used the Oracle Exadata Backup Configuration Utility, configure the pool as described in this section. For details, review the Best Practices white paper.

The system planner should consider pool protection based on the following guidelines:

- Use parity-based protection for general-purpose and capacity-optimized systems:
- * RAID-Z for protection from single-drive failure on systems subject to random workloads.

- * RAID-Z2 for protection from two-drive failure on systems with streaming workloads only.
- Use mirroring for high-performance with incrementally applied backup.
- Configure pools based on performance requirements:
- * Configure a single pool for management-optimized systems.
- * Configure two pools for performance-optimized systems. Two-pool systems should be configured by using half the drives from each tray.
- Configure log device protection:
- * Stripe log devices for RAID-Z and mirrored pool configurations.
- * Mirror log devices for RAID-Z2 pool configurations.

Note: If you used the Oracle Exadata Backup Configuration Utility, proceed to the next topic: "Configuring Oracle Exadata for a Sun ZFS Storage Appliance" on page 436.

Share Configuration

The default options for ZFS Storage Appliance shares provide a good starting point for general-purpose workloads. ZFS Storage Appliance shares can be optimized for Oracle RMAN backup and restore operations as follows:

- Create a project to store all shares related to backup and recovery of a single database. For a two-pool implementation, create two projects; one for each pool.
- Configure the shares supporting Oracle RMAN backup and restore workloads with the following values:
- * Database record size (`recordsize`): 128kB
- * Synchronous write bias (`logbias`): Throughput (for processing backup sets and image copies) or Latency (for incrementally applied backups)
- * Cache device usage (`secondary cache`): None (for backup sets) or All (when supporting incrementally applied backups or database clone operations)
- * Data compression (`compression`): Off for performance-optimized systems, LZJB or gzip-2 for capacity-optimized systems
- * Number of shares per pool: 1 for management-optimized systems, 2 or 4 for performance-optimized systems

Additional share configuration options, such as higher-level `gzip` compression or replication, can be applied to shares used to support Oracle Exadata backup and restore, as customer requirements mandate.

Customers implementing additional ZFS Storage Appliance data services should consider implementation-specific testing to verify the implications of deviations from the practices described earlier.

Configuring Oracle RMAN and the Oracle Database Instance

Oracle RMAN is an essential component for protecting the content of Oracle Exadata. Oracle RMAN can be used to create backup sets, image copies, and incrementally updated backups of Oracle Exadata content on ZFS Storage Appliances. To optimize performance of Oracle RMAN backups from Oracle Exadata to a ZFS Storage Appliance, the database administrator should apply the following best practices:

- Load balance Oracle RMAN channels evenly across the nodes of the database machine.
- Load balance Oracle RMAN channels evenly across ZFS Storage Appliance shares and controllers.

To optimize buffering of the Oracle RMAN channel to the ZFS Storage Appliance, you can tune the values of several hidden instance parameters. For Oracle Database 11*g* Release 2, the following parameters can be tuned:

- For backup and restore set:
- * _backup_disk_bufcnt=64
- * _backup_disk_bufsz=1048576
- For image copy backup and restore:
- * _backup_file_bufcnt=64
- * _backup_file_bufsz=1048576

For additional information about tuning these parameters and tuning equivalent parameters for earlier versions of the Oracle Database software, see Article ID 1072545.1: *RMAN Performance Tuning Using Buffer Memory Parameters*) at `http://support.oracle.com. (http://support.oracle.com.)`

Oracle Direct NFS (dNFS) is a high-performance NFS client that delivers exceptional performance for Oracle RMAN backup and restore operations. dNFS should be configured for customers seeking maximum throughput for backup and restore operations.

Next Steps

“Configuring Oracle Exadata for a Sun ZFS Storage Appliance” on page 436

Configuring Oracle Exadata for a Sun ZFS Storage Appliance

This section contains sample scripts showing how to attach a ZFS Storage Appliance to an Oracle Exadata. These scripts are designed to support a database named `dbname` in a one-pool and a two-pool ZFS Storage Appliance configuration.

Configure Exadata Configuring Oracle Exadata for a Sun ZFS Storage Appliance

General Implementation Steps

The implementation steps are:

1. Set up the directory structure (mount points) to mount the shares on the host.
2. Update `/etc/fstab` to mount the shares exported from the ZFS Storage Appliance to the appropriate mount points.
3. Create an `init.d` service to automate the process of mounting and unmounting the shares.
4. Update the `oranfstab` file to access the ZFS Storage Appliance exported shares or set mount on boot in `/etc/fstab`.
5. Mount the shares on the host.
6. Change the permissions of the mounted shares to match the permission settings of `ORACLE_HOME`.
7. Optionally, restart the Oracle Database instance to pick up the changes to the `oranfstab` file.

Note: If you used the Oracle Exadata Backup Configuration Utility, all steps except for step 4 and step 7 have already been performed for you. In the next section, "Detailed Implementation Steps," you may optionally perform Updating oranfstab to Access ZFS Storage Appliance Exports and step 2 of Setting the Ownership of the Mounted Shares.

Detailed Implementation Steps

Topics in this section:

- Setting Up the Directory Structure to Mount the Shares on the Host
- Updating the /etc/fstab File

- Creating an init.d Service
- Updating oranfstab to Access ZFS Storage Appliance Exports
- Mounting the Shares on the Host
- Setting the Ownership of the Mounted Shares

Setting Up the Directory Structure to Mount the Shares on the Host

Set up mount points for the shares on the host as shown:

```
mkdir -p /zfssa/dbname/backup1
mkdir -p /zfssa/dbname/backup2
mkdir -p /zfssa/dbname/backup3
mkdir -p /zfssa/dbname/backup4
```

Updating the /etc/fstab File

To update the `/etc/fstab` file, use one of the following options.

Note: The UNIX new-line escape character (\) indicates a single line of code has been wrapped to a second line in the listing below. When entering a wrapped line into `fstab`, remove the \ character and combine the two line segments, separated by a space, into a single line.

For a one-pool configuration:

```
192.168.36.200:/export/dbname/backup1 /zfssa/dbname/backup1 nfs \<br/>
 rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \<br/>   3,timeo=600 0 0
192.168.36.200:/export/dbname/backup2 /zfssa/dbname/backup2 nfs \<br/>
 rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \<br/>   3,timeo=600 0 0
192.168.36.200:/export/dbname/backup3 /zfssa/dbname/backup3 nfs \<br/>
 rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \<br/>   3,timeo=600 0 0
192.168.36.200:/export/dbname/backup4 /zfssa/dbname/backup4 nfs \<br/>
 rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \<br/>   3,timeo=600 0 0
```

For a two-pool configuration:

```
192.168.36.200:/export/dbname/backup1 /zfssa/dbname/backup1 nfs \<br/>
 rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \<br/>   3,timeo=600 0 0
192.168.36.201:/export/dbname/backup2 /zfssa/dbname/backup2 nfs \<br/>
 rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \<br/>   3,timeo=600 0 0
192.168.36.200:/export/dbname/backup3 /zfssa/dbname/backup3 nfs \<br/>
 rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \<br/>   3,timeo=600 0 0
192.168.36.201:/export/dbname/backup4 /zfssa/dbname/backup4 nfs \<br/>
 rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,nfsvers= \<br/>   3,timeo=600 0 0
```

Creating an init.d Service

Create an init.d service using the appropriate following option.

```
# !/bin/sh
#
# zfssa_dbname: Mount ZFSSA project dbname for database dbname
#
# chkconfig: 345 61 19
# description: mounts ZFS Storage Appliance shares
#


start()
{
  mount /zfssa/dbname/backup1
  mount /zfssa/dbname/backup2
  mount /zfssa/dbname/backup3
  mount /zfssa/dbname/backup4
  echo "Starting $prog: "
}


stop()
{
  umount /zfssa/dbname/backup1
  umount /zfssa/dbname/backup2
  umount /zfssa/dbname/backup3
  umount /zfssa/dbname/backup4
  echo "Stopping $prog: "
}


case "$1" in
  start)
     start
     ;;
  stop)
     stop
     ;;
  restart)
     stop
     start
     ;;
  status)
     mount
     ;;
   *)
      echo "Usage: $0 {start|stop|restart|status}"
     exit 1
esac
```

(Optional) Enable the `init.d` service for start-on-boot by entering:

```
# chkconfig zfssa_dbname on
```

(Optional) Start and stop the service manually using the service commands:

```
# service zfssa_dbname start<br/># service zfssa_dbname stop
```

Updating oranfstab to Access ZFS Storage Appliance Exports

To update the `oranfstab` file to access ZFS Storage Appliance exports, use the appropriate following option.

Note: If you used the Oracle Exadata Backup Configuration Utility, you may optionally perform this procedure.

For a one-pool configuration:

```
server: 192.168.36.200
path: 192.168.36.200
export: /export/dbname/backup1 mount: /zfssa/dbname/backup1
export: /export/dbname/backup2 mount: /zfssa/dbname/backup2
export: /export/dbname/backup3 mount: /zfssa/dbname/backup3
export: /export/dbname/backup4 mount: /zfssa/dbname/backup4
```

For a two-pool configuration:

```
server: 192.168.36.200
path: 192.168.36.200
export: /export/dbname/backup1 mount: /zfssa/dbname-2pool/backup1
export: /export/dbname/backup3 mount: /zfssa/dbname-2pool/backup3
server: 192.168.36.201
path: 192.168.36.201
export: /export/dbname/backup2 mount: /zfssa/dbname-2pool/backup2
export: /export/dbname/backup4 mount: /zfssa/dbname-2pool/backup4
```

Mounting the Shares on the Host

To mount the shares on the host, enter one of the following two options:

```
# service mount_dbname start
```

or

```
# dcli -l root -g /home/oracle/dbs_group service mount_dbname start
```

Setting the Ownership of the Mounted Shares

Change the permission settings of the mounted shares to match the permission settings of `ORACLE_HOME`. In this example, the user and group ownerships are set to `oracle:dba`.

Note: If you used the Oracle Exadata Backup Configuration Utility, you may optionally perform step 2; step 1 has already been performed for you.

1. Enter one of the following two options:<br />`# chown oracle:dba /zfssa/dbname/*`<br />or<br/>`# dcli -l root -g /home/oracle/dbs_group chown oracle:dba/zfssa/dbname/*`
2. Restart the Oracle Database instance to pick up the changes that were made to the `oranfstab` file using one of the following options:

- Restart one instance at a time (rolling upgrade), for example:
- `:$ srvctl stop instance -d dbname -i dbname1`
- `:$ srvctl start instance -d dbname -i dbname1`
- `:$ srvctl stop instance -d dbname -i dbname2`
- `:$ srvctl start instance -d dbname -i dbname2`
- `:$ srvctl stop instance -d dbname -i dbname3`
- `:$ srvctl start instance -d dbname -i dbname3`
- `:$ srvctl stop instance -d dbname -i dbname4`
- `:$ srvctl start instance -d dbname -i dbname4`
- `:$ srvctl stop instance -d dbname -i dbname5`
- `:$ srvctl start instance -d dbname -i dbname5`
- `:$ srvctl stop instance -d dbname -i dbname6`
- `:$ srvctl start instance -d dbname -i dbname6`
- `:$ srvctl stop instance -d dbname -i dbname7`
- `:$ srvctl start instance -d dbname -i dbname7`
- `:$ srvctl stop instance -d dbname -i dbname8`
- `:$ srvctl start instance -d dbname -i dbname8`
- Restart the entire database, for example:
- `:$ srvctl stop database -d dbname`
- `:$ srvctl start database -d dbname`

Oracle SPARC SuperCluster Backup

When equipped with native QDR InfiniBand and 10Gb Ethernet connectivity options, the ZFS Storage Appliance is ideal for reliably backing up the Oracle SPARC SuperCluster. Use the instructions in the following sections to configure your system:

- "Configuring the ZFS Storage Appliance for Backup"
- "Configuring Oracle SPARC SuperCluster for ZFS Storage Appliance Backup"

For detailed information on using your ZFS Storage Appliance as a backup target for Oracle SPARC SuperCluster, see the Configuring a Sun ZFS Backup Appliance with Oracle SPARC SuperCluster white paper on the NAS Storage Documentation page. Also available is an Oracle ZFS Storage ZS3-4 cluster, offered pre-racked with disk shelves as the Oracle ZFS Storage ZS3-BA to minimize set-up complexity. Integrating this appliance with Oracle SPARC SuperCluster is identical to the process described above.

Configuring the ZFS Storage Appliance for Backup

This section provides general guidelines for configuring a ZFS Storage Appliance for backup use with the Oracle SPARC SuperCluster. For detailed information, see the white paper Configuring a Sun ZFS Backup Appliance with Oracle SPARC SuperCluster on the NAS Storage Documentation page. The examples represent a ZFS Storage Appliance with two controllers (heads) and four disk shelves.

Topics in this section:

- "Configuring the ZFS Storage Appliance InfiniBand Datalinks" on page 442
- "Configuring the Oracle SPARC SuperCluster InfiniBand Switches to Add the ZFS Storage Appliance" on page 442
- "Configuring ZFS Storage Appliance Networking for Single IP Connection" on page 445
- "Configuring ZFS Storage Appliance Networking for an Active-Active Configuration" on page 446
- "Configuring the ZFS Storage Appliance Storage Pool" on page 448
- "Configuring the ZFS Storage Appliance Shares" on page 448
- "Configuring the ZFS Storage Appliance DTrace Analytics" on page 449
- "Configuring the Client NFS Mount" on page 450
- "Tuning the Solaris 11 Network and Kernel" on page 450
- "Configuring Oracle Direct NFS (dNFS)" on page 450
- "Tuning the Oracle Database Instance for Oracle RMAN Backup and Restore" on page 452
- "Creating Dedicated Services for Oracle RMAN Operations""Creating Dedicated Services for Oracle RMAN Operations" on page 454
- "Configuring Oracle RMAN" on page 454

Configuring the ZFS Storage Appliance InfiniBand Datalinks

Follow the steps in this section to configure each ZFS Storage Appliance InfiniBand connection. The eight GUIDs for the InifiniBand HBA ports that are recorded during this procedure are used to configure the Oracle SPARC SuperCluster InfiniBand switches in the next procedure.

1. Connect the ZFS Storage Appliance to the Oracle SPARC SuperCluster as described in the white paper Configuring a Sun ZFS Backup Appliance with Oracle SPARC SuperCluster on the NAS Storage Documentation page.

2. Log on to the Browser User Interface (BUI) of Head 1 and navigate to Configuration > Network.

3. Click the plus icon next to Datalinks. The Network Datalink dialogue box opens.

4. Complete the dialogue box as follows:

- Check the `IB Partition` box.
- Enter a meaningful name for the datalink name.
- Set the `Partition Key` to `8503`.
- Select Connected Mode for the Link Mode.
- Do not check the `LACP Aggregation` box.
- Select `Partition Device` `ibp0`.
- Record the GUID number (for example, `21280001ef43bb`) and click Apply.

5. Repeat steps 3 and 4 for each remaining InfiniBand interface (`ibp1`, `ibp2`, and `ibp3`).

6. Repeat steps 2 through 5 for Head 2.

Configuring the Oracle SPARC SuperCluster InfiniBand Switches to Add the ZFS Storage Appliance

In this procedure, the GUIDs of the ZFS Storage Appliance Infiniband HBA ports are added to the existing Oracle SPARC SuperCluster InfiniBand configuration. By adding these ports and using a partition key of 8503, communication between the two devices can occur.

1. Log on to the Oracle SPARC SuperCluster InfiniBand spine switch as root. By default, the spine switch is given a hostname of `<sscid>sw- ib1`, where `<sscid>` is the prefix name given to the entire Oracle SPARC SuperCluster system. In the following example, the `<sscid>` is `aiessc`.

```
login as: root
root@aiesscsw-ib1's password:
Last login: Tue Sep 25 08:19:01 2013 from dhcp-brm-bl5-204-3e
east-10-135-75-254.usdhcp.oraclecorp.com
```

2. Enter the command `enablesm` to verify that the switch is running Subnet Manager (or this command will start Subnet Manager).<br/>

```
[root@aiesscsw-ib1 ~]# enablesm
opensm (pid 15906) is already running...
Starting partitiond daemon
/usr/local/util/partitiond is already running
(You may also perform a 'restart' if wanted)
```

3. Enter the command `getmaster` to verify that this is the master switch of the configuration. If the master switch is not running on the spine switch, log out and log in to the designated master switch for the remainder of this procedure.<br/>

```
[root@aiesscsw-ib1 ~]# getmaster
Local SM enabled and running
20130913 10:16:51 Master SubnetManager on sm lid 13 sm guid
0x2128e8ac27a0a0 : SUN DCS 36P QDR aiesscsw-ib1.us.oracle.com
[root@aiesscsw-ib1 ~]#
```

4. Back up the switch configuration according the documented backup procedures (`http://docs.oracle.com/cd/E26698_01/index.html (http://docs.oracle.com/cd/E26698_01/index.html)`).

5. Enter the command `smpartition list active` to verify that partition key 0x0503 is assigned to partition name "sto" (`sto = 0x0503`).<br/> The partition key was set to 8503 on the ZFS Storage Appliance datalinks, but the InfiniBand switch reports 0503. This is intentional because the InfiniBand protocol reserves the most significant bit (0x8000) of the hexadecimal partition key (pkey) for its own use. Therefore, pkeys 0x8503 and 0x0503 are the same.<br/>

```
[root@aiesscsw-ib1 ~]# smpartition list active
# Sun DCS IB partition config file
# This file is generated, do not edit
#! version_number : 11
Default=0x7fff, ipoib : ALL_CAS=full, ALL_SWITCHES=full, SELF=
full;
SUN_DCS=0x0001, ipoib : ALL_SWITCHES=full;
ic1s10 = 0x0501,ipoib,defmember=full:
0x0021280001ef30f7,
0x0021280001ef33bf,
0x0021280001ef30b7,
0x0021280001ef314b;
ic2s10 = 0x0502,ipoib,defmember=full:
0x0021280001ef30f8,
0x0021280001ef33c0,
0x0021280001ef30b8,
0x0021280001ef314c;
sto = 0x0503,ipoib,defmember=full:
0x0021280001ef43f8,
0x0021280001ef43b7,
0x0021280001cf90c0,
0x0021280001ef43bb,
...more...
```

6. Add the ZFS Storage Appliance to the InfiniBand configuration:

- Enter the command `smpartition start` to start a reconfiguration session.<br/>

```
# smpartition start<br/>
[root@aiesscsw-ib1 ~]# smpartition start
```

- Enter the command `smpartition add` to add the eight new GUIDs to the configuration.<br/>

```
# smpartition add -n sto -port <GUID1> <GUID2> <GUID3> ...  <GUID8><br/>
[root@aiesscsw-ib1 ~]# smpartition add -n sto -port
21280001ef43bb 21280001ef43bc 21280001cf90bf 21280001cf90c0
21280001ef43f7 21280001ef43f8 21280001ef43b7 21280001ef43b8
```

- Enter the command `smpartition list modified` to verify the new GUIDs have been added correctly.<br/>

```
# smpartition list modified<br/>
[root@aiesscsw-ib1 ~]# smpartition list modified
# Sun DCS IB partition config file
<nowki># This file is generated, do not edit
#! version_number : 11
Default=0x7fff, ipoib : ALL_CAS=full, ALL_SWITCHES=full, SELF=
full;
SUN_DCS=0x0001, ipoib : ALL_SWITCHES=full;
ic1s10 = 0x0501,ipoib,defmember=full:
0x0021280001ef30f7,
0x0021280001ef33bf,
0x0021280001ef30b7,
0x0021280001ef314b;
ic2s10 = 0x0502,ipoib,defmember=full:
0x0021280001ef30f8,
0x0021280001ef33c0,
0x0021280001ef30b8,
0x0021280001ef314c;
sto = 0x0503,ipoib,defmember=full:
0x0021280001ef43f8,
0x0021280001ef43b7,
0x0021280001cf90c0,
0x0021280001ef43bb,
0x0021280001ef43bc,
0x0021280001cf90bf,
0x0021280001ef43b8,
0x0021280001ef43f7,
0x0021280001ef3048,
0x0021280001ef30af,
0x0021280001ef30f8,
0x0021280001ef30f7,
0x0021280001ef33c0,
0x0021280001ef33bf,
0x0021280001ef30cc,
0x0021280001ef342b,
0x0021280001ef30b8,
0x0021280001ef30b7,
0x0021280001ef314c,
0x0021280001ef314b,
0x0021280001efec65,
0x0021280001efec66,
0x0021280001efecb1,
0x0021280001efecb2;
```

- Enter the command `smpartition commit` to apply the new configuration and propagate configuration changes to all InfiniBand switches in the configuration.<br/>

```
# smpartition commit<br/>
[root@aiesscsw-ib1 ~]# smpartition commit
[root@aiesscsw-ib1 ~]#
```

7. Log off the InfiniBand switch.

8. Back up the InfiniBand configuration according to the documented backup procedures (`http://docs.oracle.com/cd/E26698_01/index.html (http://docs.oracle.com/cd/E26698_01/index.html)`).

Configuring ZFS Storage Appliance Networking for Single IP Connection

This configuration is only for an Oracle SPARC SuperCluster T5 with no external leaf switches. For best failover and performance, use the Active-Active Configuration (next section) for all other configurations.

Configure the ZFS Storage Appliance InfiniBand ports for network connectivity and simple cluster failover by using the following procedure to configure Port 1 with the desired IP address.

1. Log on to the BUI of Head 1 and navigate to Configuration > Network.

2. Click the plus icon next to Interfaces. The Network Interface dialogue box opens.

3. Complete the dialogue box as follows:

- Enter a meaningful name for the network interface.
- Verify that `Enable Interface` is checked.
- Verify that `Allow Administration` is checked.
- Verify that `Use IPv4 Protocol` is checked.
- Verify that the `Configure with` menu selection is `Static Address List`.
- In the box below that, enter the desired IP address with the appropriate netmask.
- Verify that `Use IPv6 Protocol` is not checked.
- Select the datalink for `ibp0` and click Apply.

4. Repeat steps 1 through 3 on Head 2 using `ibp2` as the datalink.

Configuring ZFS Storage Appliance Networking for an Active-Active Configuration

Configure the InfiniBand ports on the ZFS Storage Appliance for IP multipathing. Four IP addresses, on the private storage subnet, are needed for each ZFS Storage Appliance head (therefore, eight addresses total) because the interfaces will run in an active-active configuration.

1. Configure each InfiniBand datalink as its own network interface.

- Log on to the BUI of Head 1 and navigate to Configuration > Network.
- Click the plus icon next to Interfaces. The Network Interface dialogue box opens.
- Complete the dialogue box as follows:
- * Enter a meaningful name for the network interface.
- * Verify that `Enable Interface` is checked.
- * Verify that `Allow Administration` is checked.
- * Verify that `Use IPv4 Protocol` is checked.
- * Verify that the Configure with menu selection is Static Address List.
- * In the box below that, enter `0.0.0.0/8`.
- * Verify that `Use IPv6 Protocol` is not checked.
- * Select the datalink for `ibp0` and click Apply.
- Repeat the second and third sub-steps for the remaining datalinks (`ibp1`, `ibp2`, and `ibp3`).
- Repeat the first through fourth sub-steps on Head 2.

2. Configure the IPMP interface on Head 1.

- Log on to the BUI of Head 1 and navigate to Configuration > Network.

- Click the plus icon next to Interfaces. The Network Interface dialogue box opens.
- Complete the dialogue box as follows:
- * Enter a meaningful name for the IPMP network interface.
- * Verify that `Enable Interface` is checked.
- * Verify that `Allow Administration` is checked.
- * Verify that `Use IPv4 Protocol` is checked.
- * Verify that the `Configure with` menu selection is `Static Address List`.
- * Click the plus sign next to the empty box three times, so that four empty boxes are displayed.
- * In each empty box, enter one of the IP addresses reserved for the InfiniBand connections with its respective /24 netmask designation. As a best practice, do not use consecutive IP addresses from the block, but rather every other one (for example, all odd or all even).
- * Verify that `Use IPv6 Protocol` is not checked.
- * Check the `IP MultiPathing Group` box.
- * Check the boxes next to the interfaces corresponding to datalinks `ibp0` and `ibp3`.
- * Verify that each of the two interfaces are set to `Active` and click Apply.
- From Configuration > Network, click Routing.
- Click on the Multihoming model corresponding to `Adaptive`.

3. Configure the IPMP interface on Head 2.

- Log on to the BUI of Head 2 and navigate to Configuration > Network.
- Click the plus icon next to Interfaces. The Network Interface dialogue box opens.
- Complete the dialogue box as follows:
- * Enter a meaningful name for the IPMP network interface.
- * Verify that `Enable Interface` is checked.
- * Verify that `Allow Administration` is checked.
- * Verify that `Use IPv4 Protocol` is checked.
- * Verify that the `Configure with` menu selection is `Static Address List`.
- * Click the plus sign next to the empty box three times, so that four empty boxes are displayed.
- * In each empty box, enter one of the remaining four IP addresses reserved for the InfiniBand connections with its respective /24 netmask designation. These should be the ones not used on Head 1.
- * Verify that `Use IPv6 Protocol` is not checked.
- * Check the `IP MultiPathing Group` box.
- * Check the boxes next to the interfaces corresponding to datalinks `ibp1` and `ibp2`.
- * Verify that each of the two interfaces are set to `Active` and click Apply.
- From Configuration > Network, click Routing.
- Click on the Multihoming model corresponding to `Adaptive`.

4. Verify connectivity with the Oracle SPARC SuperCluster nodes. Verify that each node can ping each of the eight addresses used in the IPMP groups on the ZFS Storage Appliance. Add these IP addresses to the `/etc/inet/hosts` table of each node.

Configuring the ZFS Storage Appliance Storage Pool

Pool configuration assigns physical disk drive resources to logical storage pools for backup data storage. To maximize system throughput, configure two equally sized storage pools by assigning half of the physical drives in each drive tray to each storage pool.

The ZFS Storage Appliance management software presents a warning message about efficiency when two pools with the same RAID protection profile are configured. This message can be safely ignored when configuring for a high-performance Oracle RMAN backup solution.

Configuring the ZFS Storage Appliance Shares

Share configuration is the process of setting up and running NFS mount points for client access. Two projects should be created for the Oracle SPARC SuperCluster configuration: one project per pool. A project is an entity that provides a higher level management interface point for a collection of shares. To optimize share management, update the default mount point for shares contained in the project to reference the database name, such as `/export/dbname`. For a performance-optimized system, create four shares for each project in each pool, for a total of eight shares (four for each head). To configure a project, perform the following:

1. Log on to the BUI of Head 1 and navigate to Shares > Projects.

2. Click the plus icon next to Projects, enter a meaningful name for the project, and click Apply. Since a similar project will be created on the other head, uniquely name the project for Head 1, such as `H1-mydb`.

3. Click the pencil icon next to the new project name to edit the project.

4. Click General and complete the properties as follows:

- Change the `Mountpoint` to include the database name (for example, `/export/H1-mydb`).
- Change `Synchronous write bias` from `Latency` to `Throughput` and click Apply.

5. Click Protocols and add an NFS exception as follows:

- Click the plus icon next to NFS Exceptions.
- Change `Type` to `Network`.
- Enter the subnet and netmask (for example, /24) of the InfiniBand network.
- Change `Access Mode` to `Read/Write`.

- Verify that `Charset` is set to `default`.
- Check the `Root Access` box and click Apply.

6. Next to General, click Shares.

7. Create four filesystems for Head 1 and uniquely name them so they will be different from the names for Head 2. To interleave the backup streams to distribute the data across the two heads and, thereby, provide better performance, use odd-numbered names for Head 1, such as `backup1`, `backup3`, `backup5`, and `backup7`; and use even-numbered names for Head 2, such as `backup2`, `backup4`, `backup6`, and `backup8`. To create the filesystems, click the plus icon next to Filesystems, enter the name of the filesystem (`backup1`), and click Apply. Repeat this step to create the remaining three filesystems (`backup3`, `backup5`, and `backup7`).

8. Repeat steps 1 through 7 for Head 2. Remember to use a unique project name (for example, `H2-mydb`), and specify even-numbered backup IDs (`backup2`, `backup4`, `backup6`, and `backup8`) for the filesystem names.

Configuring the ZFS Storage Appliance DTrace Analytics

The ZFS Storage Appliance includes a comprehensive performance analysis tool called DTrace Analytics. DTrace Analytics is a framework that monitors important subsystem performance accounting statistics. A subset of the available accounting statistics should be monitored to provide comprehensive data on the effectiveness and performance of Oracle RMAN backup and restore workloads.

The following Analytics are available when advanced analytics are configured on the ZFS Storage Appliance (Configuration > Preferences > Enable Advanced Analytics):

- CPU: Percent utilization broken down by CPU mode
- Disk: Average number of I/O operations broken down by state of operation
- Disk: I/O bytes per second broken down by type of operation
- Disk: I/O operations per second broken down by latency
- Disk: Disks with utilization of at least 95 percent broken down by disk
- Network: Interface bytes per second broken down by direction
- Network: Interface bytes per second broken down by interface
- Protocol: NFSv3 operations per second broken down by size
- Protocol: NFSv3 operations per second broken down by type of operation
- Protocol: NFSv3 operations per second of type read broken down by latency
- Protocol: NFSv3 operations per second of type write broken down by latency
- Protocol: NFSv3 operations per second of type read broken down by size
- Protocol: NFSv3 operations per second of type write broken down by size

Implementing these accounting statistics helps end-users gain a quantitative understanding of the instantaneous and historical resource consumption and quality of service (QoS) for their specific implementation.

Configuring the Client NFS Mount

When configuring the ZFS Storage Appliance, any server that accesses the appliance, including Oracle SPARC SuperCluster nodes, is considered a client. Configuring the client NFS mount includes creating the target directory structure for access to the ZFS Storage Appliance as well as the specific NFS mount options necessary for optimal system performance. Mount options for Solaris clients are:

```
rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto=tcp,vers=3,forcedirectio
```

The mount points of the directories created on the ZFS Storage Appliance should be created on each of the Oracle SPARC SuperCluster nodes and added to their `/etc/inet/hosts` table.

Tuning the Solaris 11 Network and Kernel

The following entries should be added to the `/etc/system` file of each of Oracle SPARC SuperCluster node:

```
set rpcmod:clnt_max_conns = 8
set nfs:nfs3_bsize = 131072
```

Additionally, the following commands need to be run on each Oracle SPARC SuperCluster node every time it is rebooted:

```
/usr/sbin/ndd -set /dev/tcp tcp_max_buf 2097152
/usr/sbin/ndd -set /dev/tcp tcp_xmit_hiwat 1048576
/usr/sbin/ndd -set /dev/tcp tcp_recv_hiwat 1048576
```

Additional tuning might be necessary to achieve optimal performance. Refer to Oracle SPARC SuperCluster Tunables document 1474401.1, available at `http://support.oracle.com (http://support.oracle.com)`, for the latest information. Also, the January 2013 QFSDP release added a "ssctuner" tool that automatically sets tunables. Refer to the Oracle SPARC SuperCluster release notes for additional information.

Configuring Oracle Direct NFS (dNFS)

On each Oracle SPARC SuperCluster node, configure dNFS as follows:

1. Shut down the running instance of the Oracle Database software.

2. Change directory to `$ORACLE_HOME/rdbms/lib`.

3. Enable dNFS:<br/>

```
make -f $ORACLE_HOME/rdbms/lib/ins_rdbms.mk dnfs_on
```

4. Update the `oranfstab` file (located in `/$ORACLE_HOME/dbs`) with the server, path, and export names specific to the configuration, where:<br/>

- The server parameter refers to the local name of the ZFS Storage Appliance head on the InfiniBand network.<br/>
- The path parameters should reflect the address(es) for that head specified during configuration.<br/>
- The export parameters should reflect the mount points similar to the entries created in `/etc/vfstab`. The entries should look similar to the following.<br/>

For single IP configuration (only Oracle SPARC SuperCluster T5 without external leaf switches):

```
server: aie-zba-h1-stor
path: 192.168.30.100
export: /export/test1/backup1 mount: /zba/test1/backup1
export: /export/test1/backup3 mount: /zba/test1/backup3
export: /export/test1/backup5 mount: /zba/test1/backup5
export: /export/test1/backup7 mount: /zba/test1/backup7
server: aie-zba-h2-stor
path: 192.168.30.101
export: /export/test1/backup2 mount: /zba/test1/backup2
export: /export/test1/backup4 mount: /zba/test1/backup4
export: /export/test1/backup6 mount: /zba/test1/backup6
export: /export/test1/backup8 mount: /zba/test1/backup8<br/>
```

For IPMP Group configuration (all others):

```
server: aie-zba-h1-stor
path: 192.168.30.100
path: 192.168.30.102
path: 192.168.30.104
path: 192.168.30.106
export: /export/test1/backup1 mount: /zba/test1/backup1
export: /export/test1/backup3 mount: /zba/test1/backup3
export: /export/test1/backup5 mount: /zba/test1/backup5
export: /export/test1/backup7 mount: /zba/test1/backup7
server: aie-zba-h2-stor
path: 192.168.30.101
path: 192.168.30.103
path: 192.168.30.105
path: 192.168.30.107
```

```
export: /export/test1/backup2 mount: /zba/test1/backup2
export: /export/test1/backup4 mount: /zba/test1/backup4
export: /export/test1/backup6 mount: /zba/test1/backup6
export: /export/test1/backup8 mount: /zba/test1/backup8
```

5. Restart the Oracle Database software instance.

Tuning the Oracle Database Instance for Oracle RMAN Backup and Restore

Optimizing high-bandwidth backup and restore operations using Oracle RMAN and the ZFS Storage Appliance requires adjusting the instance parameters that control I/O buffering. For information about how to tune these parameters, see Article ID 1072545.1: RMAN Performance Tuning Using Buffer Memory Parameters) at `http://support.oracle.com. (http://support.oracle.com.)`

For Oracle SPARC SuperCluster, tuning the following four parameters should be considered:

- `_backup_disk_bufcnt` - Number of buffers used to process backup sets
- `_backup_disk_bufsz` - Size of the buffers used to process backup sets
- `_backup_file_bufcnt` - Number of buffers used to process image copies
- `_backup_file_bufsz` - Size of the buffers used to process image copies

For backup and restore operations on backup sets and image copies, set the number of buffers to 64 and the buffer size to 1 MB:

```
SQL> alter system set “_backup_disk_bufcnt”=64;
SQL> alter system set “_backup_file_bufcnt”=64;
SQL> alter system set “_backup_disk_bufsz”=1048576;
SQL> alter system set “_backup_file_bufsz”=1048576;
```

These commands may be configured persistently by adding them to the `SPFILE`, or they may be set dynamically in the Oracle RMAN run block used to execute the backup or restore operations.

The following code fragments show how to dynamically tune the buffer sizes and counts for backup and restore operations.

- Backup set backup:

```
run
{<br/>
   sql 'alter system set “_backup_disk_bufcnt”=64';<br/>
   sql 'alter system set “_backup_disk_bufsz”=1048576';<br/>
   allocate channel...
...<br/>
```

```
   backup as backupset database;
}
```

- Backup set restore:

```
run
{<br/>
   sql 'alter system set “_backup_disk_bufcnt”=64';<br/>
   sql 'alter system set “_backup_disk_bufsz”=1048576';<br/>
   allocate channel...
...<br/>
   restore database;
}
```

- Image copy backup:

```
run
{<br/>
   sql 'alter system set “_backup_file_bufcnt”=64';<br/>
   sql 'alter system set “_backup_file_bufsz”=1048576';<br/>
   allocate channel...
...<br/>
   backup as copy database;
}
```

- Image copy restore:

```
run
{<br/>
   sql 'alter system set “_backup_file_bufcnt”=64';<br/>
   sql 'alter system set “_backup_file_bufsz”=1048576';<br/>
   allocate channel...
...<br/>
   restore database;
}
```

Performing an incrementally applied backup requires reading an incremental backup set and writing to an image copy. To tune buffers for incrementally applied backups, run the following:<br/>

```
run
{<br/>
   sql 'alter system set “_backup_disk_bufcnt”=64';<br/>
   sql 'alter system set “_backup_disk_bufsz”=1048576';<br/>
   sql 'alter system set “_backup_file_bufcnt”=64';<br/>
   sql 'alter system set “_backup_file_bufsz”=1048576';<br/>
   allocate channel...
...<br/>
   recover copy of database;
}
```

Creating Dedicated Services for Oracle RMAN Operations

Two services dedicated to Oracle RMAN processing can be configured to optimize management of load balancing, high availability, and upgrades. These services can be evenly load balanced over all the nodes of an Oracle SPARC SuperCluster system. Availability and performance can be optimized by configuring the services to run on a preferred instance while preparing them to fail over to any instance in the cluster. If these services are configured, upgrading a one-quarter or one-half rack Oracle SPARC SuperCluster system does not require changing the connect string of the Oracle RMAN run block.

The `srvctl` utility is used to install services for Oracle RMAN processing. The following code fragment shows how to create two services evenly distributed over a four-node cluster that are set up to fail over to any other node in the cluster. In this example, the services are installed for a database named dbname and are named `dbname_bkup`.

```
srvctl add service -d dbname -r dbname1 -a dbname2 -s dbname_bkup1
srvctl start service -d dbname -s dbname_bkup1
srvctl add service -d dbname -r dbname2 -a dbname1 -s dbname_bkup2
srvctl start service -d dbname -s dbname_bkup2
```

Configuring Oracle RMAN

Configuring Oracle RMAN channel and parallelism includes specifying the file system targets for the Oracle RMAN backup channels and the total number of channels used for backup and restore operations. Performance benefits can be realized by configuring 16 Oracle RMAN channels spanning the available ZFS Storage Appliance shares. Configure Oracle RMAN channels such that they are evenly distributed over the Oracle Database instances and nodes in the RAC cluster and evenly distributed over the shares exported from the ZFS Storage Appliance.

The following code fragments show sample Oracle RMAN run blocks for performing backup and restore operations for backup sets and image copies as well as applying incremental merges to image copies. The sample code is based on the following database configuration:

- Database name: `dbname`
- SYSDBA login: `sys/welcome`
- Scan address: `ad01-scan`
- Service names for the backup: `dbname_bkup`

The ZFS Storage Appliance can be configured in a one-pool configuration in which the appliance exports eight shares used as eight mount points.

The Oracle RMAN run blocks for backup and restore using backup sets and image copies are shown in the examples in the sections below. In these examples, the mount points for the four-share configuration are accessed as `/zfssa/dbname/backup1` through `/zfssa/dbname/backup4`. Also, the examples are for a configuration in which the ZFS Storage Appliance exports four shares used as four mount points for 16 Oracle RMAN channels.

Backup set level 0 backup:

```
run
{<br/>
   sql ’alter system set "_backup_disk_bufcnt"=64 scope=memory’;<br/>
   sql ’alter system set "_backup_disk_bufsz"=1048576 scope=memory’;<br/>
   allocate channel ch01 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch02 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch03 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup3/%U’;<br/>
   allocate channel ch04 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch05 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch06 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch07 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup3/%U’;<br/>
   allocate channel ch08 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch09 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch10 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch11 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch12 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup3/%U’;<br/>
   allocate channel ch13 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch14 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch15 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch16 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup3/%U’;<br/>
   configure snapshot controlfile name to<br/>
   ’/zfssa/dbname/backup1/snapcf_dbname.f’;<br/>
   backup as backupset incremental level 0 section size 32g database<br/>
   tag ’FULLBACKUPSET_L0’ plus archivelog tag ’FULLBACKUPSET_L0’;
}
```

Backup set level 1 backup:

```
run
{<br/>
   sql ’alter system set "_backup_disk_bufcnt"=64 scope=memory’;<br/>
   sql ’alter system set "_backup_disk_bufsz"=1048576 scope=memory’;<br/>
   allocate channel ch01 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch02 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch03 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup3/%U’;<br/>
   allocate channel ch04 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch05 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch06 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch07 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup3/%U’;<br/>
   allocate channel ch08 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch09 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch10 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch11 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch12 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup3/%U’;<br/>
   allocate channel ch13 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch14 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch15 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch16 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup3/%U’;<br/>
   configure snapshot controlfile name to<br/>
   ’/zfssa/dbname/backup1/snapcf_dbname.f’;<br/>
   backup as backupset incremental level 1 database tag<br/>
   ’FULLBACKUPSET_L1’ plus archivelog tag ’FULLBACKUPSET_L1’;
}
```

Image copy backup:

```
run
{<br/>
   sql ’alter system set "_backup_file_bufcnt"=64 scope=memory’;<br/>
   sql ’alter system set "_backup_file_bufsz"=1048576 scope=memory’;<br/>
   allocate channel ch01 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch02 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch03 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup3/%U’;<br/>
   allocate channel ch04 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch05 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch06 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch07 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup3/%U’;<br/>
   allocate channel ch08 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch09 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch10 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch11 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch12 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup3/%U’;<br/>
   allocate channel ch13 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup2/%U’;<br/>
   allocate channel ch14 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup1/%U’;<br/>
   allocate channel ch15 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’ format ’/zfssa/dbname/backup4/%U’;<br/>
   allocate channel ch16 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’ format ’/zfssa/dbname/backup3/%U’;<br/>
   configure snapshot controlfile name to<br/>
   ’/zfssa/dbname/backup1/snapcf_dbname.f’;<br/>
   backup incremental level 1 for recover of copy with tag ’IMAGECOPY’<br/>
   database;
}
```

Incremental merge to image copy:

```
run
{<br/>
   sql ’alter system set "_backup_disk_bufcnt"=64 scope=memory’;<br/>
   sql ’alter system set "_backup_disk_bufsz"=1048576 scope=memory’;<br/>
   sql ’alter system set "_backup_file_bufcnt"=64 scope=memory’;<br/>
   sql ’alter system set "_backup_file_bufsz"=1048576 scope=memory’;<br/>
   allocate channel ch01 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch02 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch03 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch04 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch05 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch06 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch07 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch08 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch09 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch10 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch11 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch12 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch13 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch14 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch15 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch16 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   configure snapshot controlfile name to<br/>
   ’/zfssa/dbname/backup1/snapcf_dbname.f’;<br/>
   recover copy of database with tag ’IMAGECOPY’;
}
```

Restore validate:

```
run
{<br/>
   sql ’alter system set "_backup_disk_bufcnt"=64 scope=memory’;<br/>
   sql ’alter system set "_backup_disk_bufsz"=1048576 scope=memory’;<br/>
   sql ’alter system set "_backup_file_bufcnt"=64 scope=memory’;<br/>
   sql ’alter system set "_backup_file_bufsz"=1048576 scope=memory’;<br/>
   allocate channel ch01 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch02 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch03 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch04 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch05 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch06 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch07 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch08 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch09 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch10 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch11 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch12 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch13 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch14 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   allocate channel ch15 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup1’;<br/>
   allocate channel ch16 device type disk connect ’sys/welcome@ad01-<br/>
   scan/dbname_bkup2’;<br/>
   configure snapshot controlfile name to<br/>
   ’/zfssa/dbname/backup1/snapcf_dbname.f’;<br/>
   restore validate database;
}
```

Next Steps

“Oracle SPARC SuperCluster Backup” on page 440

Configuring Oracle SPARC SuperCluster for ZFS Storage Appliance Backup

This section contains sample scripts showing how to attach a ZFS Storage Appliance to an Oracle SPARC SuperCluster. These scripts are designed to support a database named `dbname` in a one-pool and a two-pool ZFS Storage Appliance configuration.

Configure SSC Configuring Oracle SPARC SuperCluster for ZFS Storage Appliance Backup

General Implementation Steps

The implementation steps are:

1. Set up the directory structure (mount points) to mount the shares on the host.
2. Update `/etc/vfstab` to mount the shares exported from the ZFS Storage Appliance to the appropriate mount points.
3. Enable the NFS client services to mount the NFS shares at reboot to automate the process of mounting and unmounting the shares.
4. Update the `oranfstab` file to access the ZFS Storage Appliance exported shares.
5. Mount the shares on the host.

6. Change the permissions of the mounted shares to match the permission settings of ORACLE_HOME.
7. Restart the Oracle Database instance to pick up the changes to the oranfstab file.

Detailed Implementation Steps

Topics in this section:

- "Setting Up the Directory Structure to Mount the Shares on the Host" on page 460
- "Updating the /etc/vfstab File" on page 460
- "Enabling the NFS Client Service" on page 461
- "Updating oranfstab to Access ZFS Storage Appliance Exports" on page 461
- "Mounting the Shares on the Host" on page 462
- "Setting the Ownership of the Mounted Shares" on page 462

Setting Up the Directory Structure to Mount the Shares on the Host

Set up mount points for the shares on the host as shown:

```
mkdir -p /zfssa/dbname/backup1
mkdir -p /zfssa/dbname/backup2
mkdir -p /zfssa/dbname/backup3
mkdir -p /zfssa/dbname/backup4
```

Updating the /etc/vfstab File

To update the /etc/vfstab file, use one of the following options.

Note: The UNIX new-line escape character (\) indicates a single line of code has been wrapped to a second line in the listing below. When entering a wrapped line into fstab, remove the \ character and combine the two line segments, separated by a space, into a single line.

For a one-pool configuration:

```
192.168.36.200:/export/dbname/backup1 - /zfssa/dbname/backup1 \<br/>
   nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= \<br/>
   tcp,vers=3,forcedirectio
192.168.36.200:/export/dbname/backup2 - /zfssa/dbname/backup2 \<br/>
   nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= \<br/>
   tcp,vers=3,forcedirectio
192.168.36.200:/export/dbname/backup3 - /zfssa/dbname/backup3 \<br/>
```

```
   nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= \<br/>
   tcp,vers=3,forcedirectio
192.168.36.200:/export/dbname/backup4 - /zfssa/dbname/backup4 \<br/>
   nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= \<br/>
   tcp,vers=3,forcedirectio
```

For a two-pool configuration:

```
192.168.36.200:/export/dbname/backup1 - /zfssa/dbname/backup1 \<br/>
   nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= \<br/>
   tcp,vers=3,forcedirectio
192.168.36.201:/export/dbname/backup2 - /zfssa/dbname/backup2 \<br/>
   nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= \<br/>
   tcp,vers=3,forcedirectio
192.168.36.200:/export/dbname/backup3 - /zfssa/dbname/backup3 \<br/>
   nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= \<br/>
   tcp,vers=3,forcedirectio
192.168.36.201:/export/dbname/backup4 - /zfssa/dbname/backup4 \<br/>
   nfs - yes rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto= \<br/>
   tcp,vers=3,forcedirectio
```

Enabling the NFS Client Service

Enable the NFS Client Service on the Solaris 11 host with the following command:

```
svcadm enable -r nfs/client
```

Updating oranfstab to Access ZFS Storage Appliance Exports

To update the `oranfstab` file to access ZFS Storage Appliance exports, use the appropriate following option.

For a one-pool configuration:

```
server: 192.168.36.200
path: 192.168.36.200
path: 192.168.36.201
path: 192.168.36.202
path: 192.168.36.203
export: /export/dbname/backup1 mount: /zfssa/dbname/backup1
export: /export/dbname/backup2 mount: /zfssa/dbname/backup2
export: /export/dbname/backup3 mount: /zfssa/dbname/backup3
export: /export/dbname/backup4 mount: /zfssa/dbname/backup4
```

For a two-pool configuration:

```
server: 192.168.36.200
path: 192.168.36.200
```

```
path: 192.168.36.202
export: /export/dbname/backup1 mount: /zfssa/dbname-2pool/backup1
export: /export/dbname/backup3 mount: /zfssa/dbname-2pool/backup3
server: 192.168.36.201
path: 192.168.36.201
path: 192.168.36.203
export: /export/dbname/backup2 mount: /zfssa/dbname-2pool/backup2
export: /export/dbname/backup4 mount: /zfssa/dbname-2pool/backup4
```

Mounting the Shares on the Host

Using the standard Solaris `mount` command, manually mount the shares:

```
# mount /zfssa/dbname/backup1
# mount /zfssa/dbname/backup2
# mount /zfssa/dbname/backup3
# mount /zfssa/dbname/backup4
```

Setting the Ownership of the Mounted Shares

Change the permission settings of the mounted shares to match the permission settings of `ORACLE_HOME`. In this example, the user and group ownerships are set to `oracle:dba`.

1. Enter:<br />`# chown oracle:dba /zfssa/dbname/*`
2. Restart the Oracle Database instance to pick up the changes that were made to the `oranfstab` file using one of the following options:

- Restart one instance at a time (rolling upgrade), for example:
- `:$ srvctl stop instance -d dbname -i dbname1`
- `:$ srvctl start instance -d dbname -i dbname1`
- `:$ srvctl stop instance -d dbname -i dbname2`
- `:$ srvctl start instance -d dbname -i dbname2`
- `:$ srvctl stop instance -d dbname -i dbname3`
- `:$ srvctl start instance -d dbname -i dbname3`
- `:$ srvctl stop instance -d dbname -i dbname4`
- `:$ srvctl start instance -d dbname -i dbname4`
- `:$ srvctl stop instance -d dbname -i dbname5`
- `:$ srvctl start instance -d dbname -i dbname5`
- `:$ srvctl stop instance -d dbname -i dbname6`
- `:$ srvctl start instance -d dbname -i dbname6`
- `:$ srvctl stop instance -d dbname -i dbname7`
- `:$ srvctl start instance -d dbname -i dbname7`

- `:$ srvctl stop instance -d dbname -i dbname8`
- `:$ srvctl start instance -d dbname -i dbname8`
- Restart the entire database, for example:
- `:$ srvctl stop database -d dbname`
- `:$ srvctl start database -d dbname`

Oracle Intelligent Storage Protocol

The Oracle Database has a layered architecture that includes the Oracle Disk Manager (ODM). The ODM provides a file management module that lets the Oracle Database use a local file system, a raw disk partition, or NFS server to store database information.

To increase database performance, the ODM interface lets the Oracle Database pass information along with each I/O request. This information defines several attributes associated with the I/O such as the file type associated with the I/O request. This lets data file and database log file writes be handled differently.

The new OISP allows the Oracle Database NFSv4 client to pass ODM optimization information to the NFSv4 server of the ZFS Storage Appliance. The ZFS Storage Appliance takes advantage of the ODM optimization information to simplify database configuration and to further increase database performance.

There are two Oracle Intelligent Storage Protocol features:

- Automatically setting the Optimal file record size for new database files
- Automatically using the optimal write bias (ZFS Latency or Throughput) for each write request

Set the Optimal file record size

The Oracle dNFS client passes the optimal record size to the ZFS Storage Appliance for each NFSv4 write request. The ZFS Storage Appliance NFSv4 server passes the record size to the ZFS file system with the I/O request. The ZFS file system then bypasses the default file system record size and uses the record size value passed with the I/O request. The record size can only be set for newly created files. If a file already exists the record size will not be changed.

Use either ZFS Latency or Throughput write mode for each request

The Oracle dNFS client passes the optimal write bias to the ZFS Storage Appliance for each NFSv4 write request. The ZFS Storage Appliance NFSv4 server passes the write bias to the

ZFS file system with the I/O request. The ZFS file system then bypasses the default file system write bias and attempts to use the write bias value passed with the I/O request. Depending on the state of the ZFS file system the write bias sent with the I/O request may be ignored.

Sun ZFS Storage Appliance Network File System Plug In for Oracle Solaris Cluster

Oracle Solaris Cluster (OCS) is a high-availability cluster software product for the Solaris Operating System.

The Sun ZFS Storage Appliance Network File System Plug In for Oracle Solaris Cluster enables OSC with the Sun ZFS Storage Appliance using NFS protocol. The plug in and readme file are available as part of the Sun ZFS Storage Appliance Network File System Plugin for Oracle Solaris Cluster on the Oracle Technology Network.

Sun ZFS Storage Appliance Plug-in for Oracle Solaris Cluster Geographic Edition

Oracle Solaris Cluster Geographic Edition software is a layered extension of the Oracle Solaris Cluster software. The Geographic Edition software protects applications from unexpected disruptions by using multiple clusters that are separated by long distances, and by using a redundant infrastructure that replicates data between these cluster sites. This plug-in coordinates data replication between remote Oracle Solaris Cluster sites using the Sun ZFS Storage Appliance remote replication service.

The plug-in package is available through the Oracle Technology Network Sun NAS Storage information page.

Sun ZFS Storage Management Plug-In for Oracle Enterprise Manager Grid Controller

The Sun ZFS Storage plug-in for Oracle Enterprise Manager Grid controller provides first-class monitoring to the grid controller environment for the Sun ZFS Storage appliance family with the ability to:

- Monitor Sun ZFS Storage appliances
- Gather storage system information, configuration information and performance information of accessible storage components

- Raise alerts and violations based on thresholds and monitoring information collected by the tool
- Provide out-of-the-box reports that complement analytics
- Support monitoring by remote agents.

Once an appliance is configured to be monitored by the grid controller, analytics worksheets and datasets are created to bridge the grid controller administrator's view to the deeper level of detail provided by the real-time analytics available within the appliance.

The management plug-in is available at the following link: Oracle Technology Network

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" on page 411 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" on page 411 page in the browser user interface.

Oracle Grid Controller Sun ZFS Storage Management Plug-In for Oracle Enterprise Manager 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 Properties" on page 134 will be created with limited access to the system, to allow the Oracle Enterprise Manager Grid Controller agent to obtain information required for monitoring, but not to make alterations to the system. An *oracle_agent* Chapter 7, "User Configuration" 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 ZFS Storage appliance.

- The Worksheet *Oracle Enterprise Manager* will be created, facilitating communication between the grid controller administrator and the storage administrator. All metrics monitored by grid controller are available from this worksheet.

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.

Oracle Virtual Machine Storage Connect Plug-in for the Sun ZFS Storage Appliance

One of many new features introduced within Oracle VM 3.0 is the Storage Connect framework. This framework enables Oracle VM 3.0 Manager to directly access storage servers and provision resources. With this framework, you can register storage servers, discover existing storage resources, create and present physical disks to server pools, and share storage repositories and Virtual Machines.

The Oracle Virtual Machine Storage Connect Plug-in for the Sun ZFS Storage Appliance is a component of the Oracle VM software suite that enables Oracle VM to provision and manage the Sun ZFS Storage Appliance for virtualization. The plug-in is installed on the Oracle VM Server(s) and communicates with the storage server(s) through workflows installed on the ZFSSA.

The plug-in and readme file are available on the Oracle Technology Network.

Sun ZFS Storage Appliance Provider For Volume Shadow Copy Service Software

Volume Shadow Copy Services (VSS) for Microsoft operating systems provides a framework to allow volume backups to be performed while applications on a system continue to write to the volumes. VSS provides a consistent interface that allows coordination between user applications that update data on disk (VSS writers) and those that back up applications (VSS requesters). Specifically, VSS provides:

- A backup infrastructure that coordinates applications with file system activities
- A location to create point in time, coalesced copies known as *shadow copies*

The Sun ZFS Storage Appliance Provider For Volume Shadow Copy Service Software is a VSS hardware provider that allows the Sun ZFS Storage Appliance to take consistent snapshots for Windows hosts which are using block targets. VSS coordinates snapshots to ensure block data is consistent. The provider communicates with a set of workflows on the appliance to coordinate taking of snapshots as seen from the application. It works over both iSCSI and Fibre Channel.

The Sun ZFS Storage Appliance Provider For Volume Shadow Copy Service Software is installed on hosts that require this functionality and coordination between applications. Complete documentation for this application integration is packaged with the downloaded components in the form of a ReadMe file. The provider software and readme file are available as part of the Sun ZFS Storage 7000 Software Providers and Plug-Ins patch on the Oracle Technology Network]. 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.

FC support with Symantec's 'DMP' / Storage Foundation

- SF - Symantec Storage Foundation 5.1
- SF HA - Storage Foundation High Availability 5.1
- SFCFS/SF Oracle RAC - Storage Foundation Cluster File System/Storage Foundation for Oracle RAC 5.1
- SFCFS/SFCFS Oracle RAC - Storage Foundation Cluster File System/Storage Foundation Cluster File System for Oracle RAC 5.1

FC support for Symantec's Storage Foundation 5.1RP2 and greater for the following OS versions

- Solaris 10 SPARC
- Solaris 10 x86
- Linux RedHat5
- Oracle Enterprise Linux (OEL)

Refer to Symantec's HCL at http://www.symantec.com/business/support/index (http://www.symantec.com/business/support/index)?page=content&id=TECH74012

Note the following restrictions:

- Symantec's "required" 7000 ASLs be installed which can be downloaded from: https://vos.symantec.com/asl
- Symantec also required SF 5.1 VM patch level of 5.1RP2 or greater which can be downloaded from: https://vos.symantec.com/patch/matrix
- Symantec also requires the following DMP parameter setting (only for 'clustered' 7000s) of:
- `:dmp_health_time=0`
- `:dmp_path_age=0`
- `:dmp_lun_retry_timeout=200`

Refer to Symantec's HW tech note which references to the 'clustered' 7000 settings: http://www.symantec.com/business/support/index (http://www.symantec.com/business/support/index)?page=content&id=TECH47728

Symantec's Storage Foundation 5.1SP2 for Windows supports FC connections to our 7000 series for the following Windows versions:

- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2

Refer to the SF 5.1SP2 HCL at http://www.symantec.com/business/support/index (http://www.symantec.com/business/support/index)?page=content&id=TECH138719

Sun ZFS Storage 7000 Storage Replication Adapter for VMware Site Recovery Manager

The Sun ZFS Storage 7000 Storage Replication Adapter (SRA) for VMware vCenter Site Recovery Manager (SRM) integrates Sun ZFS Storage 7000 appliances into VMware deployments that span multiple sites and require fast recovery in the event of a protected site service disruption. The SRA plugs into existing VMware vCenter SRM environments and allows Sun ZFS Storage 7000 appliances to be managed through VMware vCenter SRM discovery, test, and failover sequences as the recovery plan is tested and run. Usage of the SRA occurs entirely within the VMware vCenter SRM application.

The VMware administrator will need to work closely with the Sun ZFS Storage 7000 appliance administrator responsible for the appliance that hosts the VMware data stores. For further information, see the Sun ZFS Storage 7000 SRA for VMware SRM Administration Guide that is packaged in the SRA.

NOTE: The SRA can be downloaded from the Oracle Technology Network. A valid Oracle support contract for the Sun ZFS Storage 7000 appliance is required to obtain the SRA.

Index