

NearStore® VTL Version 5.2

Command Reference

Network Appliance, Inc.
495 East Java Drive,
Sunnyvale, CA 94089
USA
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Documentation comments: doccomments@netapp.com
Information Web: <http://www.netapp.com>

Part number 210-03617_A0
March 2007

Copyright and Trademark Information

Copyright Information

Copyright ©1994–2007 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted Network Appliance material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETWORK APPLIANCE “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Network Appliance reserves the right to change any products described herein at any time, and without notice. Network Appliance assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by Network Appliance. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of Network Appliance.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

NetApp, the Network Appliance logo, the bolt design, NetApp—the Network Appliance Company, DataFabric, Data ONTAP, FAServer, FilerView, FlexVol, Manage ONTAP, MultiStore, NearStore, NetCache, SecureShare, SnapDrive, SnapLock, SnapManager, SnapMirror, SnapMover, SnapRestore, SnapValidator, SnapVault, Spinnaker Networks, SpinCluster, SpinFS, SpinHA, SpinMove, SpinServer, SyncMirror, Topio, VFM, and WAFL are registered trademarks of Network Appliance, Inc. in the U.S.A. and/or other countries. Cryptainer, Cryptoshred, Datafort, and Decru are registered trademarks, and

Lifetime Key Management and OpenKey are trademarks, of Decru, a Network Appliance, Inc. company, in the U.S.A. and/or other countries. gFiler, Network Appliance, SnapCopy, Snapshot, and The evolution of storage are trademarks of Network Appliance, Inc. in the U.S.A. and/or other countries and registered trademarks in some other countries. ApplianceWatch, BareMetal, Camera-to-Viewer, ComplianceClock, ComplianceJournal, ContentDirector, ContentFabric, EdgeFiler, FlexClone, FlexShare, FPolicy, HyperSAN, InfoFabric, LockVault, NOW, NOW NetApp on the Web, ONTAPI, RAID-DP, RoboCache, RoboFiler, SecureAdmin, Serving Data by Design, SharedStorage, Simplicore, Simulate ONTAP, Smart SAN, SnapCache, SnapDirector, SnapFilter, SnapMigrator, SnapSuite, SohoFiler, SpinMirror, SpinRestore, SpinShot, SpinStor, StoreVault, vFiler, Virtual File Manager, VPolicy, and Web Filer are trademarks of Network Appliance, Inc. in the United States and other countries. NetApp Availability Assurance and NetApp ProTech Expert are service marks of Network Appliance, Inc. in the U.S.A.

Apple is a registered trademark and QuickTime is a trademark of Apple Computer, Inc. in the United States and/or other countries. Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the United States and/or other countries. RealAudio, RealNetworks, RealPlayer, RealSystem, RealText, and RealVideo are registered trademarks and RealMedia, RealProxy, and SureStream are trademarks of RealNetworks, Inc. in the United States and/or other countries.

All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

Network Appliance is a licensee of the CompactFlash and CF Logo trademarks.

Network Appliance NetCache is certified RealSystem compatible.

CONTENTS

Preface	ix
Introduction	ix
Audience	ix
Terminology.....	ix
Keyboard Conventions	ix
Typographic Conventions	ix
Special Messages	x
 1 Understanding the NearStore VTL CLI	1
Introduction	1
User Interface.....	1
Accessing the VTL CLI	2
Logging Out of the VTL CLI	3
VTL Command Structure	3
Command Prompt	3
Command Naming	4
Command Syntax	4
Command Usage	4
Positional and Named Parameters	4
Command Navigation	5
Object Identification	5
Automatic Completion.....	5
Syntax Constraints	6
Command Help Structure	6
User Interaction	7
Types of Commands.....	7
Informative Commands.....	7
Configuration Commands	9
Command Errors	10
Execution Errors.....	10
Syntax Errors	10
Using Scripts to Set Up VTL CLI Access Without Passwords	11

Installing Scripts	11
Linux and Solaris	11
Windows	12
Customizing the Script	12
Linux and Solaris	12
Windows	14
Sample Export Workflow Script	15
Running the Script	16
2 Appliance Management.....	19
Rebooting the Appliance (appliance reboot)	19
Shutting Down the Appliance (appliance shutdown)	20
Restarting the Appliance (appliance restart).....	20
Obtaining the Appliance Version (appliance version).....	20
Obtaining the Appliance Uptime (appliance uptime)	20
Showing Appliance Information (appliance show)	21
Setting the Appliance ID (appliance set)	21
Discovering Devices (appliance discover).....	22
3 Managing Configuration.....	23
Loading the Default Configuration (config load_default).....	23
4 Configuring Time	25
Showing the Appliance Time (time show).....	25
Showing the NTP Configuration (time show_ntp).....	26
Setting the NTP Configuration (time set_ntp).....	26
5 Configuring SNMP	27
Showing the SNMP Configuration (snmp show).....	27
Setting the SNMP Configuration (snmp set)	28
Sending the Test SNMP Trap (snmp sendtrap)	28
6 Configuring Network Settings	29
Showing Network Configuration (network show)	29
Setting Network Configuration (network set).....	30
7 Configuring E-Mail	31
Showing E-Mail Configuration (email show).....	31
Setting E-Mail Configuration (email set).....	32
Sending Test E-Mail (email send).....	32
8 Configuring AutoSupport	35
Showing AutoSupport Configuration (asup show).....	35
Setting AutoSupport Configuration (asup set)	36
Sending Test AutoSupport Mail (asup send)	36

9	Configuring Advanced Options	37
	Showing Advanced Options Settings (options show)	37
	Setting Advanced Options (options set)	38
10	Configuring the Administrator Account	41
	Showing Administrator Account Configuration (admin show)	41
	Setting Administrator Account Configuration (admin set)	41
	Setting the Administrator Password (admin passwd)	42
11	Configuring the Tape Operator Account	43
	Showing Tape Operator Account Configuration (tapeoperator show)	43
	Setting Tape Operator Account Configuration (tapeoperator set)	43
	Setting Tape Operator Password (tapeoperator passwd)	44
12	Configuring Debugging	45
	Listing Debug Categories (debug list)	45
	Showing Debug Categories (debug show)	45
	Enabling a Debug Category (debug enable)	46
	Disabling a Debug Category (debug disable)	46
13	Diagnosing the Appliance	49
	Showing Diagnostics (diag show)	49
	Clearing Diagnostic Messages (diag clear)	49
	Enabling Diagnostic Messages on the Console (diag enable)	50
	Disabling Diagnostic Messages on the Console (diag disable)	50
14	Showing System Diagnostic Messages.....	51
	Showing Diagnostic Messages (dmesg show)	51
15	Showing the Appliance Event Log	53
	Showing the Event Log (eventlog show)	53
16	Configuring Physical Disks	55
	Listing the Physical Disks (pdisk list)	55
	Showing Physical Disk Details (pdisk show)	56
	Discovering a Physical Disk Name or UUID (pdisk lookup)	56
17	Configuring RAID	57
	Listing Available RAID Groups (raid list)	57
	Showing Details for a RAID Group (raid show)	58
	Discovering a RAID Name or UUID (raid lookup)	59
	Showing RAID Statistics (raid stats)	59
	Removing a RAID Group (raid remove)	60
	Adding a RAID Group (raid add)	60
	Failing a Disk from a RAID Group (raid fail)	60
	Autocreating a Raid Group (raid autcreate)	61

18	Configuring Hot Spare Disks	63
	Listing Hot Spare Disks (hotspare list)	63
	Showing Hot Spare Disk Details (hotspare show)	63
	Discovering a Hot Spare Disk Name or UUID (hotspare lookup)	64
	Adding a Physical Disk to the Hot Spare Disk Pool (hotspare add)	65
	Removing a Physical Disk from Hot Spare Disk Pool (hotspare remove)	65
19	Configuring Physical Libraries	67
	Listing Available Physical Libraries (plib list)	67
	Showing Physical Library Details (plib show)	68
	Discovering a Physical Library Name or UUID (plib lookup)	68
	Listing Inventory Details (plib inventory)	69
	Adding a Physical Library (plib add)	69
	Removing a Physical Library (plib remove)	70
	Exporting a Volume (plib move_to_eep)	70
	Canceling an Export Request (plib eep_cancel)	70
	Polling EEP (plib eep_check)	71
	Rescanning Inventory (plib rescan)	71
20	Configuring Physical Tape Drives	73
	Listing Physical Tape Drives (pdrive list)	73
	Showing Physical Tape Drive Details (pdrive show)	74
	Discovering a Physical Tape Drive Name or UUID (pdrive lookup)	74
	Showing I/O Statistics for Physical Tape Drive (pdrive stats)	75
	Adding a Physical Tape Drive (pdrive add)	75
	Removing a Physical Tape Drive (pdrive remove)	75
	Enabling a Physical Tape Drive (pdrive enable)	76
	Disabling a Physical Tape Drive (pdrive disable)	76
	Listing Physical Tape Drive Content (pdrive contents)	76
21	Configuring Customer Information	79
	Listing Customers (customer list)	79
	Showing Customer Details (customer show)	80
	Discovering a Customer Name or UUID (customer lookup)	80
	Removing a Customer (customer remove)	81
	Adding a New Customer (customer add)	81
	Updating Customer Information (customer update)	82
	Setting the Password for a Customer (customer passwd)	82
22	Configuring Storage Pools	85
	Listing Names of Storage Pools (sp list)	85
	Showing Storage Pool Details (sp show)	86
	Discovering a Storage a Pool Name or UUID (sp lookup)	86
	Adding a Storage Pool to the Appliance (sp add)	87
	Removing a Storage Pool from the Appliance (sp remove)	87

Updating Storage Pool Information (sp update)	88
23 Configuring Virtual Libraries	89
Listing Virtual Libraries (vlib list)	89
Showing Virtual Library Details (vlib show)	90
Discovering a Virtual Library Name or UUID (vlib lookup)	91
Creating a Virtual Library (vlib add)	91
Removing a Virtual Library (vlib remove)	93
Updating Virtual Library Information (vlib update)	93
Importing a Volume into a Virtual Library (vlib import)	94
Exporting a Volume from a Virtual Library (vlib export)	95
Listing Inventory Details of a Virtual Library (vlib inventory)	95
24 Configuring Virtual Tapes	97
Listing Virtual Tapes (vtape list)	97
Showing Virtual Tape Details (vtape show)	98
Discovering a Virtual Tape Name or UUID (vtape lookup)	99
Creating a Virtual Tape (vtape add)	99
Removing a Virtual Tape (vtape remove)	100
Updating a Virtual Tape (vtape update)	100
Moving a Volume from One Virtual Library to Another (vtape move)	101
Loading a Volume into a Virtual Library Drive (vtape load)	101
Unloading a Volume from a Virtual Library Drive (vtape unload)	101
25 Configuring Virtual Tape Drive Type	103
Listing a Virtual Tape Drive Type (vtdt list)	103
Showing Virtual Tape Drive Type Details (vtdt show)	104
Discovering a Virtual Tape Drive Type Name or UUID (vtdt lookup)	104
Adding a New Virtual Tape Drive Type (vtdt add)	105
Removing a Virtual Tape Drive Type (vtdt remove)	105
Updating a Virtual Tape Drive Type (vtdt update)	106
26 Configuring Hosts	107
Listing Host Names (host list)	107
Showing Host Details (host show)	108
Discovering a Host Name or UUID (host lookup)	108
Adding a New Host (host add)	109
Removing a Host (host remove)	109
Updating a Host (host update)	109
27 Configuring Ports	111
Listing Ports (port list)	111
Showing Port Details (port show)	111
Discovering a Port Name or UUID (port lookup)	112
28 Showing Appliance Actions	113

	Showing Action Details (action show).....	113
29	Importing or Exporting Requests	115
	Showing Import or Export Request Details (iereq show)	115
	Requesting the Loading of a Volume (iereq load).....	116
	Canceling a Request (iereq cancel)	116
	Requeuing a Request (iereq requeue)	116
30	Using the Pick List.....	119
	Showing the Pick List (picklist show).....	119
	Notifying That a Volume Has Been Found (picklist found).....	119
	Notifying That a Volume Is Missing (picklist missing).....	120
31	Configuring Host or Port Access	121
	Listing Host Access (access host_list)	121
	Defining Scope for Host Access (access host_scope).....	122
	Adding Host Access (access host_add).....	122
	Removing Host Access (access host_remove).....	122
	Listing Port Access (access port_list)	123
	Defining Scope for Port Access (access port_scope).....	123
	Adding Port Access (access port_add).....	124
	Removing Port Access (access port_remove).....	124
32	Configuring Virtual Tape Drive	125
	Listing Virtual Tape Drives (vdrive list).....	125
	Showing Virtual Tape Drive Details (vdrive show)	125
	Showing I/O Statistics for a Virtual Tape Drive (vdrive stats).....	126
33	Using Shadow Tape APIs.....	127
	Listing Shadow Tapes (stape list)	127
	Showing Shadow Tape Details (stape show)	127
	Discovering a Shadow Tape Name or UUID (stape lookup).....	128
34	Configuring ACSLS	129
	Enabling ACSLS.....	129
	Setting the ACSLS Server (acsls set).....	129
	Showing ACSLS Status (acsls show)	130
	Listing Barcode Ranges (acsls list_barcode_range).....	131
	Adding Barcode Ranges to VTL (acsls add_barcode_range).....	131
	Removing Barcode Ranges to VTL (acsls remove_barcode_range).....	132
35	Using ACS Libraries	133
	Listing ACS Libraries (acs list).....	133
	Showing ACS Libraries (acs list_acs).....	134
	Listing ACS Libraries LSM Details (acs list_lsm).....	134
	Listing ACS Library Drive Details (acs list_drive).....	135

Listing ACS Libraries CAP Details (acs list_cap)	136
Adding ACS Libraries (acs add)	136
Removing ACS Libraries (acs remove)	137
Listing Inventory Details of an ACS (acs inventory)	137
Making a Batch CAP Request (acs batch_cap)	138
Listing Batch CAP Requests (acs list_batch_cap)	138
Canceling a Batch CAP Request (acs cancel_batch_cap)	139
Moving Tapes to CAP (acs eject)	139
36 Licensing	141
Listing Installed Licenses (license list)	141
Showing Installed Licenses (license show)	141
Adding a License (license add)	142
37 Upgrading Firmware	143
Installing VTL Firmware (appliance install)	144
Listing Shelf Controller Firmware (shelffirmware list)	145
Installing Downloaded Shelf Controller Firmware on Specific Shelves (shelffirmware install)	145
Installing Downloaded Shelf Controller Firmware on All Shelves (shelffirmware install image_name all)	145
Installing Prepackaged Shelf Controller Firmware (shelffirmware install all)	146
Viewing Shelf Controller Firmware Status (shelffirmware status)	146
Listing Disk Firmware (diskfirmware list)	147
Installing Disk Firmware on Disks of a Specific Model (diskfirmware install)	147
Installing Downloaded Firmware on All Disks (diskfirmware install all ftp)	148
Installing Prepackaged Firmware on All Disks (diskfirmware install all default)	148
Viewing Disk Firmware Status (diskfirmware status)	148

Preface

Introduction

This guide describes the NearStore® VTL command line interface (CLI). For information about NearStore VTL appliance features and functions, see the *NearStore VTL System Administration Guide*.

Audience

This guide is designed for a qualified system administrator. It is assumed that this person is familiar with storage area networks (SANs) and backup applications.

Terminology

This guide uses the term *type* to mean pressing one or more keys on the keyboard. It uses the term *enter* to mean pressing one or more keys and then pressing the Enter key.

Keyboard Conventions

When describing key combinations, this guide uses the hyphen (-) to separate individual keys. For example, Ctrl-D means pressing the Control and D keys simultaneously. Also, this guide uses the term *Enter* to refer to the key that generates a carriage return, although the key is named *Return* on some keyboards.

Typographic Conventions

The following table describes typographic conventions used in this guide.

Convention	Type of information
<i>Italic font</i>	Words or characters that require special attention. Placeholders for information you must supply. For example, if the guide says to enter the <code>arp -d host_name</code> command, you enter the characters arp -d followed by the actual name of the host.

Monospaced font	Command and daemon names. Information displayed on the system console or other computer monitors. The contents of files.
Bold monospaced font	Words or characters you type. What you type is always shown in lowercase letters, unless you must type it in uppercase letters.
<Text within angle brackets>	Placeholder text that describes command output. The text in the angle brackets appears in the interface as a label for information that is displayed as a result of the successful execution of the command.

Special Messages

This guide contains a special message that is described as follows:

Note _____

A note contains important information that helps you install or operate the storage system efficiently.

Understanding the NearStore VTL CLI

This chapter provides an overview of the NearStore Virtual Tape Library (VTL) CLI. It contains the following subsections:

- “Introduction,” page 1
- “User Interface,” page 1
- “User Interaction,” page 7
- “Using Scripts to Set Up VTL CLI Access Without Passwords,” page 11

Introduction

The command line interface (CLI) provides a scriptable and consistent way for users to control the VTL. The CLI allows users to view and modify VTL objects and take advantage of the services provided by them.

User Interface

The User Interface section consists of the following subsections:

- “Accessing the VTL CLI,” page 2
- “Command Prompt,” page 3
- “Logging Out of the VTL CLI,” page 3
- “VTL Command Structure,” page 3
- “Command Usage,” page 4
- “Syntax Constraints,” page 6
- “Command Help Structure,” page 6

Accessing the VTL CLI

You should run Secure Shell (SSH) software on a server or desktop computer and use it to access the VTL.

The VTL supports two methods of issuing commands through the CLI: by issuing commands from the interactive command line after logging in or by issuing a command directly from the SSH command line.

With the first method of issuing commands, you begin by entering only your user name and the IP or host name of the VTL. The system prompts you for your password and then displays the default prompt, at which you enter commands. The following text shows the correct format and an example:

```
[host prompt] $ ssh user_name@IP_address_or_host_name

[host prompt] $ ssh admin@10.72.185.22
admin@10.72.185.22's password:*****
NearStore VTL CLI
User admin logged in.
VTL>
```

The second method of issuing commands allows you to enter a command along with your user name and IP address or host name. The system prompts you for your password and then immediately executes the command, bypassing the default prompt. The following text shows the correct format for this method and an example:

```
[host prompt] $ ssh user_name@IP_address_or_host_name command

[host prompt] $ ssh admin@10.72.185.22 network show
admin@10.72.185.22's password:
Network Configuration:-

                                Host Name: VTL_deux22
                                IP Address: 10.72.185.22
                                Netmask: 255.255.255.0
                                Gateway Address: 10.72.185.1
                                Primary DNS Server: 10.72.185.45
                                Secondary DNS Server: 10.54.134.25

[host prompt] $ echo $?
0
```

The 0 in the last line of the preceding text indicates that the command executed successfully.

Note

The VTL CLI returns proper exit values on execution of a command:

- 0– Command success
 - 1– Command syntax error
 - 2– Command execution error
-

The following text shows an example of a login command that contains a command syntax error and the returned error message:

```
[host prompt] $ ssh admin@10.72.185.22 vlib export
admin@10.72.185.22's password:*****
Error: Insufficient Parameters
Export a Volume from a Virtual Library.
Usage: vlib export vtape_id
[host prompt] $ echo $?
1
```

The following text shows an example of a login command that contains an execution error and the returned error message:

```
[host prompt] $ ssh admin@10.72.185.22 customer remove admin
admin@10.72.185.22's password:*****
Error: Customer option is not enabled.
[host prompt] $ echo $?
2
```

Logging Out of the VTL CLI

To log out of the CLI session, enter `exit` or `quit` at the prompt:

```
VTL> exit
```

Note

The login credentials used in an SSH session to the VTL expire after one hour of inactivity. Although the SSH connection might still be active, you need to open a new SSH session to revalidate the credentials and log in again to the VTL.

VTL Command Structure

The VTL command structure section consists of the following subsections:

- “Command Prompt,” page 3
- “Command Naming,” page 4
- “Command Syntax,” page 4

Command Prompt

The VTL CLI provides a default prompt:

```
VTL>
```

Command Naming

Users control the VTL by treating it as a group of objects and the services that the objects provide. Each of these objects and services has a unique name, as follows:

- Objects: *ptape*, *customer*, *host*, *ldisk*, *vtldt*, *vlib*, *vtape*, *vlt*, *pdisk*, and so on
- Services: *add*, *remove*, *list*, *set*, *get*, *load*, *unload*, *export*, and so on

Command Syntax

Commands take the following format:

```
VTL> object service argument(s)
```

Arguments can be in positional parameter format or in named format. These formats are discussed next.

Command Usage

The Command Usage section consists of the following subsections:

- “Positional and Named Parameters,” page 4
- “Command Navigation,” page 5
- “Object Identification,” page 5
- “Automatic Completion,” page 5

Positional and Named Parameters

The VTL CLI supports arguments in positional parameter format and named parameter format. Most of the arguments shown in this guide are in positional parameter format.

The following shows an argument in positional parameter format:

```
param_value
```

The following shows an argument in named parameter format:

```
param_name=param_value
```

The following text displays two examples of correct command syntax. The first shows arguments in positional parameter format and the second shows arguments in named parameter format.

```
VTL> customer add name email
```

```
VTL> snmp set station=station_ip community=community version=version
```

Any positional parameter can be named, as shown:

```
VTL> raid remove raid_name
```

```
VTL> raid remove name=disk_name
```

Combinations of positional and named parameters also are allowed, as shown:

```
VTL> snmp set station_ip community=community version=version
```

```
VTL> snmp set station_ip community version=version
```

In the case of named parameters, the order of the parameters does not matter, as shown:

```
VTL> snmp set station=station_ip community=community version=version
```

```
VTL> snmp set community=community version=version station=station_ip
```

Command Navigation

The CLI command history to enable the user to navigate through the previous commands:

```
VTL> display history
```

Object Identification

All operations support UUID-based and name-based formats for the identification of a requisite object, as shown:

```
VTL> customer remove customer_uuid
```

```
VTL> customer remove customer_name
```

Automatic Completion

The VTL CLI supports automatic completion of commands, subcommands, and parameters.

To display a list of all commands in columns, press Tab at the command prompt:

```
VTL>
```

access	admin	customer	dmesg	host
acs	appliance	debug	email	hotspare
acsls	asup	diag	eventlog	iereq
action	config	display	help	license ...

To display a list of commands beginning with the letter *v*, type *v* and press Tab at the command prompt:

```
VTL> v
```

```
vdrive vlib vtape vtdt
```

To display a list of subcommands for a command, type a command and press Tab at the command prompt:

```
VTL> vtape
```

```
add help list load lookup move remove unload update
```

To display a list of arguments for a command, enter the command and subcommand and press Tab at the command prompt:

```
VTL> vtape update
enable_disk_compression    enable_tape_smart_sizing    export_needed ...
```

Syntax Constraints

Named parameters always follow positional parameters; the VTL ignores any unnamed parameters that follow named parameters.

The following examples show *incorrect* commands:

```
VTL> snmp set station=station_ip community version
VTL> snmp set station_ip community=community version
VTL> snmp set station=station_ip community=community version
```

The following examples show *correct* commands:

```
VTL> snmp set station_ip community version
VTL> snmp set station_ip community version=version
VTL> snmp set station_ip community=community version=version
VTL> snmp set station=station_ip com=community version=version
```

Parameters containing spaces must be within quotes, as shown:

```
VLT> sp add name description="Default Storage Pool"
```

The following Boolean parameters are valid:

- true or false
- yes or no
- enable or disable

The case is not relevant, and any substring also is valid.

If the user enters only the name of a Boolean parameter, the system takes its value as true.

```
VTL> options set customer=1
VTL> options set customer=y
VTL> options set customer=NO
```

Command Help Structure

The Help system is designed to guide users in selecting a command to perform the desired task. For example, to view a list of possible arguments for a given command, type the command name, a space, and a question

mark, and press Enter. The help system lists a set of keywords associated with the command or displays a short description of the applicable arguments. Commands without parameters display usage information.

Command	Description
help	Lists all available commands.
help customer	Lists all the services provided by customer object.
help customer add	Displays a short description of command and parameters.
customer	Lists all the services provided by customer object.
customer <help>	Lists all the services provided by customer object.
customer add	Displays a short description of command and parameters.

User Interaction

The User Interaction section consists of the following subsections:

- “Types of Commands,” page 7
- “Command Errors,” page 10

Types of Commands

Command output depends on the nature of the service and result of the command’s execution. This section consists of the following subsections:

- “Informative Commands,” page 7
- “Configuration Commands,” page 9

Informative Commands

Informative commands provide information about an object’s state or configuration.

They list objects:

```
VTL> customer list
Admin
John
```

They list objects based on a given range:

```
VTL> vtape list [index] [count]
VTL> vtape list index=10 count=5
vlib1_vol11
vlib1_vol12
```

```
vlib1_vol13  
vlib1_vol14  
vlib1_vol15
```

Note

- Both `index` and `count` are optional parameters.
 - The default value of `index` is 0.
 - The default value of `count` is all entries.
-

Use informative commands to obtain information about objects. You can specify that the output appear in any of the following formats:

- XML format
- Simple name format
- Parsable name format
- Tabular format

The following is an example of XML format:

```
VTL> display format xml  
VTL> customer show John  
<customer> <name>John</name> <id>104</id> <contact>John Smith</contact>  
<day_phone>12345678</day_phone> <night_phone>87654321</night_phone>  
<mobile_phone>9988776655</mobile_phone>  
<pager>5566778899</pager></customer>
```

The following is an example of simple name format (with a value pair per line):

```
VTL> display format simple  
VTL> customer show John  
Name: John  
Customer ID: 104  
Allocated Capacity: 5 GB  
Used Capacity: 1.2 GB  
Contact Name: John Smith  
E-Mail ID: johns@netapp.com  
Day Phone: 12345678  
Night Phone: 87654321  
Mobile: 9988776655  
Pager: 5566778899
```

The following is an example of parsable name format (with value pairs separated by the *or* symbol (`()`) for scripting purposes):

```
VTL> display format parsable
```

```
VTL> customer show John
```

```
Name: John|Customer ID: 104|Allocated Capacity: 5 GB|Used Capacity:
1.2 GB|Contact Name: John Smith|E-Mail ID: johnx@netapp.com|Day Phone:
12345678|Night Phone: 87654321| Mobile: 9988776655|Pager: 5566778899
```

The following is an example of tabular format:

```
VTL> display format tabular
```

```
VTL> customer show John
```

Name	CustomerID	E-Mail ID	DayPhone	NightPhone	Mobile
John	104	john@abc.com	12345678	87654321	1234
Tom	58	tom@xyz.com	11223344	55667788	4321

When you want to use a `display format` command to control the output of another command, you can enter the commands separately, as shown in the previous examples. However, as an alternative, you can enter both commands at the same time, provided that they are separated by a semicolon, as shown:

```
VTL> display format parsable;customer show John
```

```
Name: John|Customer ID: 104|Allocated Capacity: 5 GB|Used Capacity:
1.2 GB|Contact Name: John Smith|E-Mail ID: johnx@netapp.com|Day Phone:
12345678|Night Phone: 87654321| Mobile: 9988776655|Pager: 5566778899
```

Configuration Commands

Use configuration commands to create objects, remove objects, or update existing objects in the appliance.

The following is an example of how to create an object:

```
VTL> customer add John passwd 456 "John S"
```

The following is an example of how to remove an object:

```
VTL> customer remove John
```

The following is an example of how to update object information:

```
VTL> customer update John customer_id
```

Note

The user can specify only the fields to be updated.

The following is an example of how to set the password:

```
VTL> customer set_passwd
```

```
Enter new password>*****
```

```
Confirm password>*****
```

Note

The password should not be cleartext.

Command Errors

In case of command error, the system displays a message, as shown:

```
Error: Error Message
```

Note

Error: precedes all error messages.

This section consists of the following subsections:

- “Execution Errors,” page 10
- “Syntax Errors,” page 10

Execution Errors

Execution errors are described, as shown:

```
VTL> customer remove ABCD  
Error: Customer not found
```

Syntax Errors

Syntax errors are displayed with a usage message, as in the following examples.

This example shows the result of a request for an object or service that does not exist:

```
VTL> virtual list  
Error: Invalid Command
```

This example shows the result of a user providing fewer parameters than required:

```
VTL> sp add SP1  
Error: Insufficient Parameters  
Add a Storage Pool to the Appliance.  
Usage: sp add name description
```

This example shows the result of a user providing superfluous parameters:

```
VTL> sp add SP1 "new storage pool" temp  
Error: Superfluous Parameters  
Add a Storage Pool to the Appliance.
```

Usage: sp add name description

This example shows the result of a user providing two or more values for the same parameter:

```
VTL> snmp set station=1.2.3.4 station=4.5.6.7
Error: Duplicate Params
Update SNMP Configuration.
Usage: snmp set [station] [community] [version] [enable]
```

This example shows the result of a user providing a parameter that is not supported:

```
VTL> snmp set station=1.2.3.4 client=5.6.7.8
Error: Invalid Parameters
Update SNMP Configuration.
Usage: snmp set [station] [community] [version] [enable]
```

This example shows the result of a user providing a parameter value that is invalid:

```
VTL> snmp set station=snmp_server
Error: Invalid Values assigned to Parameters
Update SNMP Configuration.
Usage: snmp set [station] [community] [version] [enable]
```

Using Scripts to Set Up VTL CLI Access Without Passwords

This section describes the installation and use of scripts to set up access to the VTL CLI without passwords. It consists of the following subsections:

- “Installing Scripts,” page 11
- “Customizing the Script,” page 12
- “Sample Export Workflow Script,” page 15
- “Running the Script,” page 16

Installing Scripts

You can install scripts to operate in either the Linux and Solaris environments or in the Windows environment.

Linux and Solaris

Check to see if an Expect package and SSH software are already installed on your system. If not, check to see if they are on the Linux or Solaris installation CD or if they are available from the distributor. (Installing the Expect package also should install the TCL package that is a required dependency.)

Windows

- Install the Expect package for Windows. The package can be downloaded from <http://bmrc.berkeley.edu/people/chaffee/expectnt.html>.
- Install the OpenSSH client package for Windows. The package can be downloaded from <http://sshwindows.sourceforge.net/download/>.

Customizing the Script

In the Expect script, change the path to the Expect and SSH programs based on where the Expect package was installed earlier. This section consists of the following subsections:

- “Linux and Solaris,” page 12
- “Windows,” page 14

Linux and Solaris

To modify the script to suit the needs of your environment, change the first line in the script (`#!/usr/local/bin/expect`) to something like `#!/path/to/expect`.

Similarly, change the line `set ssh /usr/bin/ssh` to something like `set ssh /path/to/ssh`.

The following text is a sample script for a Linux or Solaris environment:

```
#!/usr/local/bin/expect

set ssh /usr/bin/ssh

set timeout -1
set username "admin"
set password "NetApp"
set command "help"

if {$argc < 1 || $argc > 4} {
    puts "Usage: $argv0 <vtl> \["command"\] \["password"\] \["username"\]"
    puts "ssh \["username"\]:\["password"\]@<vtl> \["command"\]"
    puts "Defaults: "
    puts "  command = $command"
    puts "  password = $password"
    puts "  username = $username"
    exit 1
}

set vtl [lindex $argv 0]
```

```
if {[string length [lindex $argv 1]] != 0} {
    set command [lindex $argv 1]
}

if {[string length [lindex $argv 2]] != 0} {
    set password [lindex $argv 2]
}

if {[string length [lindex $argv 3]] != 0} {
    set username [lindex $argv 3]
}

spawn $ssh -o NumberOfPasswordPrompts=1 $username@$vtl $command

while {1} {
    expect {
        -ex "ssh: $vtl: Name or service not known" {
            expect eof
            exit 1
        } -ex "Are you sure you want to continue connecting (yes/no)?" {
            send -- "yes\r"
        } -ex "$username@$vtl's password:" {
            send -- "$password\r"
            break
        }
    }
}

while {1} {
    expect {
        -ex "Permission denied." {
            expect eof
            exit 2
        } -re "Error: " {
            expect eof
            exit 100
        } eof {
            exit 0
        }
    }
}
```

Windows

To modify the script to suit the needs of your environment, change the line `set ssh "C:\\Program Files\\OpenSSH\\bin\\ssh.exe"` to `set ssh "C:\\Path To\\OpenSSH\\bin\\ssh.exe"` depending on where the OpenSSH client was installed earlier.

The following text is a sample script for the Windows platform:

```
set ssh "C:\\Program Files\\OpenSSH\\bin\\ssh.exe"

set timeout -1
set username "admin"
set password "NetApp"
set command "help"

if {$argc < 1 || $argc > 4} {
    puts "Usage: $argv0 <vtl> \\[\"command\\\"] \\[password\\] \\[username\\]"
    puts "ssh \\[username\\]:\\[password\\]@<vtl> \\[\"command\\\"]"
    puts "Defaults: "
    puts "  command = $command"
    puts "  password = $password"
    puts "  username = $username"
    exit 1
}

set vtl [lindex $argv 0]

if {[string length [lindex $argv 1]] != 0} {
    set command [lindex $argv 1]
}

if {[string length [lindex $argv 2]] != 0} {
    set password [lindex $argv 2]
}

if {[string length [lindex $argv 3]] != 0} {
    set username [lindex $argv 3]
}

spawn $ssh -o NumberOfPasswordPrompts=1 $username@$vtl $command

while {1} {
    expect {
        -ex "ssh: $vtl: Name or service not known" {
            expect eof
            exit 1
        }
    }
}
```

```

    } -ex "Are you sure you want to continue connecting (yes/no)?" {
        send -- "yes\r"
    } -ex "$username@$vtl's password:" {
        send -- "$password\r"
        break
    }
}

while {1} {
    expect {
        -ex "Permission denied." {
            expect eof
            exit 2
        } -re "Error: " {
            expect eof
            exit 100
        } eof {
            exit 0
        }
    }
}

```

Sample Export Workflow Script

```

#!/bin/sh
POLL_INTERVAL=60

if [ $# -lt 2 ]
then
    echo 1>&2 "Usage: $0 <vtl> <vtape> [interval]"
    echo 1>&2 "Exports <vtape> from <vtl>.  Polls once every \
[interval] seconds for completion"
    echo 1>&2 "Defaults:"
    echo 1>&2 "    interval = 60"
    exit 1
fi

VTL=$1
NO_PASSWD_CMD="./unix.exp $VTL"
VTAPE=$2

if [ -n $3 ]
then

```

```
POLL_INTERVAL=$3
fi

$NO_PASSWD_CMD "vlib export $VTAPE"
if [ $? -ne 0 ]
then
    echo "Error exporting $VTAPE from $VTL"
    exit 1
fi

while sleep $POLL_INTERVAL
do
    $NO_PASSWD_CMD "iereq show" | grep "State: transferring data"
    if [ $? -eq 1 ]
    then
        echo "Export of $VTAPE from $VTL is complete"
        # email the admin that the physical tape is ready to
        # be collected.
        break
    fi
done

exit 0
```

Running the Script

If you are working in a Linux or Solaris environment, run the script by entering the following command:

```
$ ./unix.exp <vtl> ["command"] [password] [username]
```

The username, password, and command values default to admin, NetApp, and help, respectively.

If you are working in a Windows environment, run the script by entering the following command:

```
C:\Scripts> "C:\Expect-5.2\bin\expect.exe" windows.exp <vtl> <command>
```

If the command syntax includes spaces (for example, `network show` or `vlib list`), type the command within double quotes, as shown:

```
"network show" / "vlib list")
```

The resulting output can be piped to other common utilities, such as `grep`, `sed`, `cut`, and further processed.

If the command is successful, the script exits with an exit status of 0. (This can be checked by using the `$?` variable in Linux and Solaris.) One of the following non-zero values is returned if an error occurs:

- If the VTL host is not reachable, the exit status is 1.
- If the password is incorrect, the exit status is 2.

- If there is an error in the command invocation, the exit status is 100.

2

Appliance Management

This chapter describes the commands that you use to manage the NearStore VTL appliance through the CLI. It consists of the following sections:

- “Rebooting the Appliance (appliance reboot),” page 19
- “Shutting Down the Appliance (appliance shutdown),” page 20
- “Restarting the Appliance (appliance restart),” page 20
- “Obtaining the Appliance Version (appliance version),” page 20
- “Obtaining the Appliance Uptime (appliance uptime),” page 20
- “Showing Appliance Information (appliance show),” page 21
- “Setting the Appliance ID (appliance set),” page 21
- “Discovering Devices (appliance discover),” page 22

Rebooting the Appliance (appliance reboot)

SYNOPSIS

```
appliance reboot [wait_time]
```

ARGUMENTS

<code>wait_time</code>	Number of seconds after which the appliance is rebooted
------------------------	---

RESULT

A successful `appliance reboot` command has an exit status of 0 and results in a warm (or soft) reboot of the appliance. The default is no wait. On failure, `appliance reboot` returns a non-zero exit status and relevant error message.

Shutting Down the Appliance (appliance shutdown)

SYNOPSIS

```
appliance shutdown [wait_time]
```

ARGUMENTS

<i>wait_time</i>	Number of seconds after which the appliance is shut down
------------------	--

RESULT

A successful `appliance shutdown` command shuts down the appliance. The default is `no wait`. On failure, `appliance shutdown` returns a non-zero exit status and relevant error message.

Restarting the Appliance (appliance restart)

SYNOPSIS

```
appliance restart
```

ARGUMENTS

None

RESULT

A successful `appliance restart` command has an exit status of 0 and restarts the VTL process. On failure, `appliance restart` returns a non-zero exit status and relevant error message.

Obtaining the Appliance Version (appliance version)

SYNOPSIS

```
appliance version
```

ARGUMENTS

None

RESULT

A successful `appliance version` command displays the version of the appliance being used. On failure, `appliance version` returns a non-zero exit status and relevant error message.

Obtaining the Appliance Uptime (appliance uptime)

SYNOPSIS

appliance uptime

ARGUMENTS

None

RESULT

A successful `appliance uptime` command outputs the cumulative uptime of the appliance since it was initialized. On failure, `appliance uptime` returns a non-zero exit status and relevant error message.

Showing Appliance Information (appliance show)

SYNOPSIS

appliance show

ARGUMENTS

None

RESULT

A successful `appliance show` command returns the following output:

```
<State> <Total Capacity> <Used Capacity> <Allocated Capacity>  
<Appliance_ID>
```

On failure, `appliance show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> appliance show
```

Monitoring Information:-

```
                State: WARNING  
Total Capacity: 501865250816  
Used Capacity: 0  
Allocated Capacity: 0  
Appliance ID: VTL_BTC01
```

Setting the Appliance ID (appliance set)

SYNOPSIS

appliance set appliance_id

ARGUMENTS

<code>appliance_id</code>	New ID to be used as <code>appliance_id</code>
---------------------------	--

RESULT

A successful `appliance_set` command sets the appliance ID to the specified value. On failure, `appliance_set` returns a non-zero exit status and relevant error message.

Discovering Devices (`appliance discover`)

SYNOPSIS

```
appliance discover
```

ARGUMENTS

None

RESULT

A successful `appliance discover` command triggers discovery of all the physical devices connected to the VTL, such as physical tape drives, physical libraries, and physical disks. On failure, `appliance discover` returns a non-zero exit status and relevant error message.

3

Managing Configuration

This chapter describes the CLI command that you use to load the default configuration of the NearStore VTL appliance.

Loading the Default Configuration (config load_default)

SYNOPSIS

```
config load_default
```

ARGUMENTS

None

RESULT

A successful `config load_default` command loads the default configuration, overwriting any existing configuration, and automatically reboots the reconfigured system. On failure, `config load_default` returns a non-zero exit status and relevant error message.

4

Configuring Time

This chapter describes the CLI commands that you use to configure time for the NearStore VTL. It consists of the following sections:

- “Showing the Appliance Time (time show),” page 25
- “Showing the NTP Configuration (time show_ntp),” page 26
- “Setting the NTP Configuration (time set_ntp),” page 26

Showing the Appliance Time (time show)

SYNOPSIS

```
time show
```

ARGUMENTS

None

RESULT

A successful `time show` command displays the current appliance time. A failure displays an error message.

EXAMPLE

```
VTL> time show
```

```
Appliance:-
```

```
Current Time: Tue Nov 21 19:58:04 2006
```

Showing the NTP Configuration (time show_ntp)

SYNOPSIS

```
time show_ntp
```

ARGUMENTS

None

RESULT

A successful `time show_ntp` command displays the NTP Server IP address and enable status. A failure prints an error message.

EXAMPLE

```
VTL> time show_ntp
NTP Server Configuration:-
    IP Address: 192.168.1.2
    Enabled: False
```

Setting the NTP Configuration (time set_ntp)

SYNOPSIS

```
time set_ntp [ipaddr] [enable]
```

ARGUMENTS

<i>ipaddr</i>	NTP server IP address
<i>enable</i>	Flag to enable or disable the NTP server

RESULT

A successful `time set_ntp` command sets the NTP server IP address. If the *ipaddr* argument is invalid, the command returns a non-zero exit status and relevant error message.

Configuring SNMP

This chapter describes the commands that you use to configure SNMP for the NearStore VTL appliance. It consists of the following sections:

- “Showing the SNMP Configuration (snmp show),” page 27
- “Setting the SNMP Configuration (snmp set),” page 28
- “Sending the Test SNMP Trap (snmp sendtrap),” page 28

Showing the SNMP Configuration (snmp show)

SYNOPSIS

```
snmp show
```

ARGUMENTS

None

RESULT

A successful `snmp show` command displays the SNMP configuration, in the following format:

```
<Management Station IP Address> <Community> <Traps Version> <Enabled Status>
```

On failure, `snmp show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> snmp show
```

```
SNMP Configuration:-
```

```
    Management Station IP Address: 10.72.185.30
```

```
        Community: public
```

```
        Traps Version: 2
```

```
        Enabled: True
```

Setting the SNMP Configuration (snmp set)

SYNOPSIS

```
snmp set [station] [community] [version] [enable=on|off]
```

ARGUMENTS

<i>station</i>	IP address of the SNMP management station
<i>community</i>	Name of the SNMP community
<i>version</i>	SNMP version to be used
<i>enable=on off</i>	Flag to enable or disable SNMP configurations

RESULT

A successful `snmp set` command configures the SNMP with user-specified values. If `enable` is passed as an argument, `snmp set` enables or disables the SNMP notifications accordingly. If the type or format of the arguments is invalid, `snmp set` returns a non-zero exit status and relevant error message.

Sending the Test SNMP Trap (snmp sendtrap)

SYNOPSIS

```
snmp sendtrap
```

ARGUMENTS

None

RESULT

A successful `snmp sendtrap` command sends a sample trap to the configured SNMP server. On failure, `snmp sendtrap` returns a non-zero exit status and relevant error message.

6

Configuring Network Settings

This chapter describes the network-configuration commands that you use for the NearStore VTL appliance. It consists of the following sections:

- “Showing Network Configuration (network show),” page 29
- “Setting Network Configuration (network set),” page 30

Showing Network Configuration (network show)

SYNOPSIS

```
network show
```

ARGUMENTS

None

RESULT

A successful `network show` command has the exit status 0 and displays network configuration details, in the following format:

```
<Host Name> <IP Address> <Netmask> <Gateway> <Primary DNS Server>  
<Secondary DNS Server>
```

On failure, `network show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> network show
```

```
Network Configuration:-
```

```
Host Name: VTL_BTC_01  
IP Address: 10.72.185.37  
Netmask: 255.255.255.0
```

Gateway Address: 10.72.185.1
Primary DNS Server: 10.72.34.45
Secondary DNS Server: 10.45.24.35

Setting Network Configuration (network set)

SYNOPSIS

`network set [ipaddr] [netmask] [gateway] [pri_dns] [sec_dns] [host name]`

ARGUMENTS

<i>ipaddr</i>	IP address of the VTL appliance
<i>netmask</i>	Netmask IP address for your network
<i>gateway</i>	IP address of the gateway for the default route
<i>pri_dns</i>	IP address of primary DNS
<i>sec_dns</i>	IP address of secondary DNS
<i>host name</i>	Host name of the VTL appliance

Note

Initially you need to specify all the arguments. Thereafter you can have single or multiple arguments.

RESULT

A successful `network set` command sets the network configuration with user-specified values. If the arguments passed are invalid, `network set` returns a non-zero exit status and relevant error message.

Configuring E-Mail

This chapter describes the commands that you use to configure e-mail for the NearStore VTL appliance. It consists of the following sections:

- “Showing E-Mail Configuration (email show),” page 31
- “Setting E-Mail Configuration (email set),” page 32
- “Sending Test E-Mail (email send),” page 32

Showing E-Mail Configuration (email show)

SYNOPSIS

```
email show
```

ARGUMENTS

None

RESULT

A successful `email show` command displays e-mail configuration details, in the following format:

```
<E-Mail Server IP Address> <Admin E-Mail Address (from)>  
<Recipients E-Mail List (to)> <Recipients E-Mail List (cc)>  
<Enable Status>
```

On failure, `email show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> email show
```

```
EMail Configuration:-
```

```
    E-Mail Server IP Address: 10.72.156.34  
    Admin E-Mail Address (from): admin_vtl@netapp.com
```

```
Recipients E-Mail List (to): abc@netapp.com
Recipients E-Mail List (cc): xyz@netapp.com
Enabled: False
```

Setting E-Mail Configuration (email set)

SYNOPSIS

```
email set [smtp_addr] [from] [to] [cc] [enable=on|off]
```

ARGUMENTS

<i>smtp_addr</i>	SMTP server IP address
<i>from</i>	VTL administrator e-mail address
<i>to</i>	E-mail address to which NearStore VTL will send e-mail notifications
<i>cc</i>	E-mail address to which NearStore VTL will send a copy of the e-mail notifications
<i>enable=on off</i>	Flag to enable or disable e-mail notifications

Note

For initial configuration, all the parameters are required except *cc*.

RESULT

A successful `email set` command has the exit status 0 and sets the e-mail configuration with the user-specified values. If the parameter is `enable=on`, the `email set` command enables e-mail notifications and sets the e-mail enable state to True. If the parameter is `enable=off`, the `email set` command disables e-mail notifications and sets the e-mail enable state to False. If the arguments specified are invalid, `email set` returns a non-zero exit status and relevant error message.

Sending Test E-Mail (email send)

SYNOPSIS

```
email send
```

ARGUMENTS

None

RESULT

A successful `email send` command sends a sample e-mail to the recipient's e-mail address. On failure, `email send` returns a non-zero exit status and relevant error message. An action is logged if the mail server configuration results in failure of `email send`.

Configuring AutoSupport

This chapter describes the commands that you use to configure AutoSupport for the NearStore VTL. It contains the following sections:

- “Showing AutoSupport Configuration (asup show),” page 35
- “Setting AutoSupport Configuration (asup set),” page 36
- “Sending Test AutoSupport Mail (asup send),” page 36

Showing AutoSupport Configuration (asup show)

SYNOPSIS

```
asup show
```

ARGUMENTS

None

RESULT

A successful `asup show` command displays AutoSupport configuration details, in the following format:

```
<Recipients E-Mail List (to)> <Recipients E-Mail List (cc)>  
<Enable Status>
```

On failure, `asup show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> asup show
```

```
Auto Support Configuration:-
```

```
    Recipients E-Mail List (to): autosupport@netapp.com
```

```
    Recipients E-Mail List (cc):
```

```
        Enabled: False
```

Setting AutoSupport Configuration (asup set)

SYNOPSIS

```
asup set [from] [to] [cc] [enable=on|off]
```

ARGUMENTS

<i>from</i>	VTL administrator e-mail address
<i>to</i>	E-mail address to which NearStore VTL sends AutoSupport notifications
<i>cc</i>	E-mail address to which NearStore VTL sends a copy of the AutoSupport notifications
<i>enable=on off</i>	Flag to enable or disable AutoSupport notifications

RESULT

A successful `asup set` command has the exit status 0 and sets the AutoSupport configuration with the user-specified values. If the parameter is `enable=on`, the `asup set` command enables ASUP notifications and sets the ASUP enable state to True. If the parameter is `enable=off`, the `asup set` command disables ASUP notifications and sets the ASUP enable state to False. If the arguments specified are invalid, `asup set` returns a non-zero exit status and relevant error message.

Sending Test AutoSupport Mail (asup send)

SYNOPSIS

```
asup send
```

ARGUMENTS

None

RESULT

A successful `asup send` command sends an AutoSupport Notification to the recipient's e-mail address. On failure, `asup send` returns a non-zero exit status and relevant error message. An action is logged if the mail server configuration results in failure of `asup send`.

Configuring Advanced Options

This chapter describes the commands that you use to configure advanced options for the NearStore VTL appliance. It consists of the following sections:

- “Showing Advanced Options Settings (options show),” page 37
- “Setting Advanced Options (options set),” page 38

Showing Advanced Options Settings (options show)

SYNOPSIS

`options show`

ARGUMENTS

None

RESULT

A successful `options show` command displays the advanced options status as true or false, in the following format:

```
<Use Customer> <Use Storage Pools> <Enable Physical Tape Validation>
<Use Tape Pick List> <Enable Host Access Control>
<Enable Port Assignment> <Enable Volume Set Addressing>
<Enable Shadow Tape> <Enable Over Allocation> <Use Acsls>
<Enable EEP> <Automatically Move Physical Tapes From EEP To Slots>
<Move Physical Tapes To EEP After Cloning>
<Move Physical Tapes To EEP After Virtual Export> <Enable Deferred Cloning>
```

Otherwise, `options show` returns a non-zero exit status and relevant error message.

EXAMPLE

VTL> **options show**

Advanced Options:-

```

                                Use Customer: False
                                Use Storage Pools: False
    Enable Physical Tape Validation: True
                                Use Tape Pick List: False
                                Enable Host Access Control: True
                                Enable Port Assignment: True
    Enable Volume Set Addressing: False
                                Enable Shadow Tape: False
                                Enable Over Allocation: True
                                Use Acsls: False
                                Enable EEP: False
    Move Physical Tapes From EEP To Slots: False
    Move Physical Tapes To EEP After Cloning: False
    Move Physical Tapes To EEP After Export: False
                                Enable Deferred Cloning: False
```

Setting Advanced Options (options set)

SYNOPSIS

```
options set [customer] [sp] [tape_validation] [tape_pick_list]
[host_access] [port_mapping] [volume_set_addressing] [enable_shadow_tape]
[over_allocation] [enable_backend_0d] [enable_eep]
[enable_auto_move_from_eep_to_slots] [enable_move_to_eep_after_cloning]
[enable_move_to_eep_after_export] [deferred_cloning] [use_acsls]
```

ARGUMENTS

<i>customer</i>	Provides the functionality to set up and administer multiple customer accounts.
<i>sp</i>	Gives you the ability to group physical disks into mutually exclusive storage resources.
<i>tape_validation</i>	Turns on physical tape validation for the appliance.
<i>tape_pick_list</i>	Each time the backup application requests a tape (barcode) that is not in a virtual library, NearStore VTL generates an action. This action requests that you load the physical tape, identified by a barcode, into an available physical tape drive.
<i>host_access</i>	Enables and disables device access on a host-by-host basis.

<code>port_mapping</code>	Provides the ability to make virtual devices available only on specific front-end ports.
<code>volume_set_addressing</code>	Turn on this feature if you are using an operating system that requires volume set addressing.
<code>enable_shadow_tape</code>	Enables the creation of shadow tapes.
<code>over_allocation</code>	When you enable this feature, NearStore VTL allocates disk space to virtual tapes as needed, regardless of how many tapes you initially create.
<code>enable_backend_0d</code>	Uses Port 0d for backend devices. (VTL700 only)
<code>enable_eep</code>	With EEP enabled, the NearStore VTL can automatically manage the physical tape library entry/exit port to move physical tapes to the port, and to move tapes from the port to appropriate slots.
<code>enable_auto_move_from_eep_to_slots</code>	If enabled, the NearStore VTL automatically moves physical tapes from the entry/exit port to a slot.
<code>enable_move_to_eep_after_cloning</code>	If enabled, after the NearStore VTL finishes cloning a virtual tape to a physical tape, the physical tape is automatically moved from the library tape drive to the library entry/exit port.
<code>enable_move_to_eep_after_export</code>	If enabled, after the NearStore VTL finishes exporting a virtual tape to a physical tape, the physical tape is automatically moved from the library tape drive to the library entry/exit port.
<code>deferred_cloning</code>	If enabled, the span of time during which the NearStore VTL will not create clones. Any changes that are made to the virtual tapes during this time will be held in queue until the deferred cloning time period has expired.
<code>use_acsls</code>	Use ACSLS to control physical libraries.

RESULT

A successful `options set` command sets options to user-specified values. If the arguments are invalid, `options set` returns a non-zero exit status and relevant error message.

Configuring the Administrator Account

This chapter describes commands that you use to configure the administrator account for the NearStore VTL appliance. It consists of the following sections:

- “Showing Administrator Account Configuration (admin show),” page 41
- “Setting Administrator Account Configuration (admin set),” page 41
- “Setting the Administrator Password (admin passwd),” page 42

Showing Administrator Account Configuration (admin show)

SYNOPSIS

```
admin show
```

ARGUMENTS

None

RESULT

A successful `admin show` command displays the administrator login ID in the format `<Login Id>`. On failure, `admin show` returns a non-zero exit status and relevant error message.

Setting Administrator Account Configuration (admin set)

SYNOPSIS

```
admin set login_id
```

ARGUMENTS

<code>login_id</code>	Identification (name) with which a user can log in to the appliance with administrator privileges
-----------------------	---

RESULT

A successful `admin set` command sets the administrator user name and password to the user-specified values. If the arguments passed are invalid, `admin set` returns a non-zero exit status and relevant error message.

Setting the Administrator Password (`admin passwd`)

SYNOPSIS

`admin passwd`

ARGUMENTS

None

RESULT

A successful `admin passwd` command sets the administrator password to the new password specified by the user. On failure, `admin passwd` returns a non-zero exit status and relevant error message.

EXAMPLE

The `set passwd` command is interactive. It asks the user for the old password, the new password, and the new password once more to avoid typing errors.

```
VTL> admin passwd
Old Password:*****
Enter the Password:*****
Re-enter the Password:*****
```

Configuring the Tape Operator Account

This chapter describes the commands that you use to configure the tape operator account for the NearStore VTL appliance. It consists of the following sections:

- “Showing Tape Operator Account Configuration (tapeoperator show),” page 43
- “Setting Tape Operator Account Configuration (tapeoperator set),” page 43
- “Setting Tape Operator Password (tapeoperator passwd),” page 44

Showing Tape Operator Account Configuration (tapeoperator show)

SYNOPSIS

```
tapeoperator show
```

ARGUMENTS

None

RESULT

A successful `tapeoperator show` command displays tape operator login name in the format `<Login Id>`. On failure, `tapeoperator show` returns a non-zero exit status and relevant error message.

Setting Tape Operator Account Configuration (tapeoperator set)

SYNOPSIS

```
tapeoperator set login_id
```

ARGUMENTS

<code>login_id</code>	Identification (name) with which a user can log in to the appliance with tape operator privileges
-----------------------	---

RESULT

A successful `tapeoperator set` command sets the tape operator user name and password to the user-specified values. If the arguments passed are invalid, `tapeoperator set` returns a non-zero exit status and relevant error message.

Setting Tape Operator Password (`tapeoperator passwd`)

SYNOPSIS

```
tapeoperator passwd
```

ARGUMENTS

None

RESULT

A successful `tapeoperator passwd` command sets the tape operator password to the new password specified by the user. On failure, `tapeoperator passwd` returns a non-zero exit status and relevant error message.

EXAMPLE

The command `tapeoperator passwd` is interactive. It asks for old password, the new password, and the new password once more to avoid typing errors.

```
VTL> tapeoperator passwd
Old Password:*****
Enter the Password:*****
Re-enter the Password:*****
```

Configuring Debugging

This chapter describes the commands that you use to configure debugging for the NearStore VTL appliance. It consists of the following sections:

- “Listing Debug Categories (debug list),” page 45
- “Showing Debug Categories (debug show),” page 45
- “Enabling a Debug Category (debug enable),” page 46
- “Disabling a Debug Category (debug disable),” page 46

Listing Debug Categories (debug list)

SYNOPSIS

```
debug list [category] [index] [count]
```

ARGUMENTS

<i>category</i>	Debug category to be listed
<i>index</i>	Index from which the listing should begin
<i>count</i>	Number of entries to be listed

RESULT

A successful `debug list` command lists all the categories of debug. If the argument is invalid, `debug list` returns a non-zero exit status and relevant error message.

Showing Debug Categories (debug show)

SYNOPSIS

```
debug show [category] [index] [count]
```

ARGUMENTS

<i>category</i>	Category for which details are to be shown
<i>index</i>	Index from which the listing should begin
<i>count</i>	Number of entries to be listed

RESULT

A successful `debug show` command prints details for all of the categories, in the following format:

```
<Name> <Enabled> <Description>
```

If *category* is passed as an argument, details for that category are displayed. On failure, `debug show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> debug show cli_server
```

```
1. Debug Category:-
```

```
                Name: cli_server
                Enabled: False
                Description: CLI Server
```

Enabling a Debug Category (debug enable)

SYNOPSIS

```
debug enable category
```

ARGUMENTS

<i>category</i>	Category that has to be enabled
-----------------	---------------------------------

RESULT

A successful `debug enable` command enables the specified debug category. If the *category* passed as the argument is invalid, `debug enable` returns a non-zero exit status and relevant error message.

Disabling a Debug Category (debug disable)

SYNOPSIS

```
debug disable category
```

ARGUMENTS

<i>category</i>	Category that has to be disabled
-----------------	----------------------------------

RESULT

A successful `debug disable` command disables the specified debug category. If the *category* passed as the argument is invalid, `debug disable` returns a non-zero exit status and relevant error message.

Diagnosing the Appliance

This chapter describes the commands that you use to diagnose the NearStore VTL appliance. It consists of the following sections:

- “Showing Diagnostics (diag show),” page 49
- “Clearing Diagnostic Messages (diag clear),” page 49
- “Enabling Diagnostic Messages on the Console (diag enable),” page 50
- “Disabling Diagnostic Messages on the Console (diag disable),” page 50

Showing Diagnostics (diag show)

SYNOPSIS

```
diag show
```

ARGUMENTS

None

RESULT

A successful `diag show` command displays diagnostics messages. On failure, `diag show` returns a non-zero exit status and relevant error message.

Clearing Diagnostic Messages (diag clear)

SYNOPSIS

```
diag clear
```

ARGUMENTS

None

RESULT

A successful `diag clear` command clears the stored diagnostic messages. On failure, `diag clear` returns a non-zero exit status and relevant error message.

Enabling Diagnostic Messages on the Console (`diag enable`)

SYNOPSIS

`diag enable`

ARGUMENTS

None

RESULT

A successful `diag enable` command starts displaying diagnostic messages on the VTL console. On failure, an error message is displayed.

Disabling Diagnostic Messages on the Console (`diag disable`)

SYNOPSIS

`diag disable`

ARGUMENTS

None

RESULT

A successful `diag disable` command stops displaying diagnostic messages on the VTL console. On failure, an error message is displayed.

Showing System Diagnostic Messages

This chapter describes the command that you use to show diagnostic messages for the NearStore VTL appliance.

Showing Diagnostic Messages (`dmesg show`)

SYNOPSIS

```
dmesg show
```

ARGUMENTS

None

RESULT

A successful `dmesg show` command displays diagnostics messages. On failure, `dmesg show` returns a non-zero exit status and relevant error message.

Showing the Appliance Event Log

This chapter describes the command that you use to show the event log for the NearStore VTL appliance.

Showing the Event Log (eventlog show)

SYNOPSIS

```
eventlog show
```

ARGUMENTS

None

RESULT

A successful `eventlog show` command displays events. On failure, `eventlog show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> eventlog show
```

```
1. Event:-
```

```
Event Code: 104  
Creation Time: Tue Nov 21 17:59:54 2006  
Description: Export of tape vlib1_vol107 \  
from library vlib1 added to pick list.&nbsp; Action 1100 issued
```


Configuring Physical Disks

This chapter describes the commands that you use to configure the physical disks for the NearStore VTL appliance. It consists of the following sections:

- “Listing the Physical Disks (pdisk list),” page 55
- “Showing Physical Disk Details (pdisk show),” page 56
- “Discovering a Physical Disk Name or UUID (pdisk lookup),” page 56

Note ---

The physical disk commands refer to *available* disks. A disk is available to the VTL if it is not a spare, is not in the maintenance pool, and does not belong to a RAID group.

Listing the Physical Disks (pdisk list)

SYNOPSIS

```
pdisk list [pdisk_id]
```

ARGUMENTS

<i>pdisk_id</i>	Physical disk identifier (UUID or serial number)
-----------------	--

RESULT

A successful `pdisk list` command displays a list of physical disks available at the back end. If *pdisk_id* is invalid, `pdisk list` returns a non-zero exit status and relevant error message.

Showing Physical Disk Details (pdisk show)

SYNOPSIS

`pdisk show [pdisk_id]`

ARGUMENTS

<code>pdisk_id</code>	Physical disk identifier (UUID or serial number)
-----------------------	--

RESULT

A successful `pdisk show` command displays details of all the available physical disks. If `pdisk_id` is specified, details of that physical disk are printed. The output format is:

<Port Name> <Shelf> <Bay> <Serial No> <Total Capacity>

If the `pdisk_id` specified is invalid, `pdisk show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> pdisk show
1. Physical Disk:-
                        Port Name: 0b
                        Shelf: 2
                        Bay: 3
                        Serial No: Y66E75GE
                        Total Capacity: 250933083136
```

Discovering a Physical Disk Name or UUID (pdisk lookup)

SYNOPSIS

`pdisk lookup pdisk_id|pdisk_uuid`

ARGUMENTS

<code>pdisk_id</code>	Physical disk identifier (serial number)
<code>pdisk_uuid</code>	Physical disk identifier (UUID)

RESULT

If the argument is `pdisk_id`, the `pdisk lookup` command displays the UUID of the corresponding available physical disk. If the argument is `pdisk_uuid`, the `pdisk lookup` command displays the physical disk serial number of the corresponding physical disk. If the argument passed is invalid, `pdisk lookup` returns a non-zero exit status and relevant error message.

Configuring RAID

This chapter describes the commands that you use to configure RAID for the NearStore VTL appliance. It consists of the following sections:

- “Listing Available RAID Groups (raid list),” page 57
- “Showing Details for a RAID Group (raid show),” page 58
- “Discovering a RAID Name or UUID (raid lookup),” page 59
- “Showing RAID Statistics (raid stats),” page 59
- “Removing a RAID Group (raid remove),” page 60
- “Adding a RAID Group (raid add),” page 60
- “Failing a Disk from a RAID Group (raid fail),” page 60
- “Autocreating a Raid Group (raid autcreate),” page 61

Listing Available RAID Groups (raid list)

SYNOPSIS

```
raid list [raid_name]
```

ARGUMENTS

<i>raid_name</i>	Name of the RAID group
------------------	------------------------

RESULT

A successful `raid list` command displays the list of available RAID groups. On failure, `raid list` returns a non-zero exit status and relevant error message.

Showing Details for a RAID Group (raid show)

SYNOPSIS

`raid show [raid_name]`

ARGUMENTS

<code>raid_name</code>	Name of the RAID group (name or UUID)
------------------------	---------------------------------------

RESULT

A successful `raid show` command displays details for each disk in the RAID group specified in the argument, in the following format:

<Name> <Storage Pool> <Total Capacity> <Allocated Capacity> <PDisk>
<Port Name> <Shelf> <Bay> <Serial No>

If no argument is specified, details are printed for all the RAID groups available. If the argument specified is invalid, `raid show` returns a non-zero exit status and relevant error message.

EXAMPLE

VTL> **raid show**

1. Raid:-

```

                Name: LD0001
        Storage Pool: Default Storage Pool
        Total Capacity: 501865250816
    Allocated Capacity: 10738466816
        Raid Status: Normal
                PDisk: 1
        Port Name: 0b
                Shelf: 2
                Bay: 0
        Serial No: Y66E7DAE
                PDisk: 2
        Port Name: 0b
                Shelf: 2
                Bay: 1
        Serial No: Y66EHC7E
                PDisk: 3
        Port Name: 0b
                Shelf: 2
                Bay: 2
        Serial No: Y66CDTJE
```

Discovering a RAID Name or UUID (raid lookup)

SYNOPSIS

```
raid lookup raid_id|raid_uuid
```

ARGUMENTS

<i>raid_id</i>	Name assigned to the RAID group (label)
<i>raid_uuid</i>	UUID of the RAID group

RESULT

A successful `raid lookup` command prints the RAID label if the input argument is *raid_uuid*. It prints the UUID if the argument is *raid_id*. If the argument is invalid, `raid lookup` returns a non-zero exit status and relevant error message.

Showing RAID Statistics (raid stats)

SYNOPSIS

```
raid stats [raid_id]
```

ARGUMENTS

<i>raid_id</i>	Label assigned to the RAID group (name or UUID)
----------------	---

RESULT

A successful `raid stats` command displays the I/O statistics for the RAID specified as the argument. If no argument is specified, statistics for all the RAID groups are printed. If the argument is invalid, `raid stats` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> raid stats
```

```
1. Physical Disk I/O Statistics:-
```

```

                                Name: LD0001
                        Total Capacity: 501865250816
                Allocated Capacity: 15991832576
                System Overhead: 10737942528
                        IO/Sec: 0
                        MB/Sec: 0.000000
                        Errors/Sec: 0
                #Active Streams: 0
                        Usage State: UNUSED
```

Removing a RAID Group (raid remove)

SYNOPSIS

```
raid remove raid_id
```

ARGUMENTS

<i>raid_id</i>	Label assigned to the RAID group (name or UUID)
----------------	---

RESULT

If the RAID group specified is not in use, `raid remove` removes it from the available RAID groups. If the *raid_id* specified is invalid, `raid remove` returns a non-zero exit status and relevant error message.

Adding a RAID Group (raid add)

SYNOPSIS

```
raid add name disk_count disk_id1 disk_id2 disk_id3 [disk_id4]  
[disk_id5] [disk_id6] [disk_id7] [sp_id]
```

ARGUMENTS

<i>name</i>	Name of the RAID group
<i>disk_count</i>	Number of disks
<i>disk_id#</i>	Name or UUID of the physical disks that will be part of the RAID group
<i>sp_id</i>	Storage pool to which RAID group belongs

RESULT

A successful `raid add` command creates a RAID group that has the name specified in the argument and contains the disks specified in the argument list. If *name*, *disk_id#*, or *sp_id* is invalid, `raid add` returns a non-zero exit status and relevant error message.

Failing a Disk from a RAID Group (raid fail)

SYNOPSIS

```
raid fail raid_id old_disk_id [new_disk_id]
```

ARGUMENTS

<i>raid_id</i>	Label assigned to the RAID group (name or UUID)
<i>old_disk_id</i>	ID of the disk to be replaced
<i>new_disk_id</i>	ID of the replacement disk

RESULT

A successful `raid fail` command replaces the existing disk with the new disk specified as the argument. If a new disk is not specified, the disk is replaced with a disk in the spare disk pool. On failure, `raid fail` returns a non-zero exit status and relevant error message.

Autocreating a Raid Group (`raid autocreate`)

SYNOPSIS

```
raid autocreate
```

ARGUMENTS

None

RESULT

A successful `raid autocreate` command creates a RAID group with these specifications:

- All available disks are formatted to create RAID group and hot spare disks.
- Disks of equal size are used.
- A RAID group consists of five to seven disks.
- A hot spare disk is added every 42nd disk.
- If possible, disks are allocated across shelves.

On failure, `raid autocreate` returns a non-zero exit status and relevant error message.

Configuring Hot Spare Disks

This chapter describes the commands that you use to configure hot spare disks for the NearStore VTL appliance. It consists of the following sections:

- “Listing Hot Spare Disks (hotspare list),” page 63
- “Showing Hot Spare Disk Details (hotspare show),” page 63
- “Discovering a Hot Spare Disk Name or UUID (hotspare lookup),” page 64
- “Adding a Physical Disk to the Hot Spare Disk Pool (hotspare add),” page 65
- “Removing a Physical Disk from Hot Spare Disk Pool (hotspare remove),” page 65

Listing Hot Spare Disks (hotspare list)

SYNOPSIS

```
hotspare list [pdisk_id]
```

ARGUMENTS

<i>pdisk_id</i>	Physical disk identifier (UUID or serial number)
-----------------	--

RESULT

A successful `hotspare list` command displays a list of physical disks reserved as hot spare replacements. On failure, `hotspare list` returns a non-zero exit status and relevant error message.

Showing Hot Spare Disk Details (hotspare show)

SYNOPSIS

```
hotspare show [pdisk_id]
```

ARGUMENTS

<i>pdisk_id</i>	Physical disk identifier (UUID or serial number)
-----------------	--

RESULT

A successful `hotspare show` command displays details for each of the physical disks that are in the hot spare pool, in the following format:

<Storage Pool> <Port Name> <Shelf> <Bay> <Serial No> <Total Capacity>

If the *pdisk_id* is specified, details are printed only for that physical disk. If the specified *pdisk_id* is invalid, `hotspare show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> hotspare show
1. HotSpare:-
           Storage Pool: Default Storage Pool
           Port Name: 0b
           Shelf: 2
           Bay: 3
           Serial No: Y66E75GE
           Total Capacity: 250933083136
```

Discovering a Hot Spare Disk Name or UUID (hotspare lookup)

SYNOPSIS

```
hotspare lookup pdisk_id|pdisk_uuid
```

ARGUMENTS

<i>pdisk_id</i>	Physical disk identifier (serial number)
<i>pdisk_uuid</i>	Physical disk identifier (UUID)

RESULT

If the argument is *pdisk_id*, the `hotspare lookup` command displays the UUID of the corresponding physical disk. If the argument is *pdisk_uuid*, the `hotspare lookup` command displays the serial number of the corresponding physical disk. If the argument passed is invalid, `hotspare lookup` returns a non-zero exit status and relevant error message.

Adding a Physical Disk to the Hot Spare Disk Pool (hotspare add)

SYNOPSIS

```
hotspare add pdisk_id [sp_id]
```

ARGUMENTS

<i>pdisk_id</i>	Physical disk identifier (UUID or serial number)
<i>sp_id</i>	Storage pool identifier

RESULT

A successful `hotspare add` command adds the physical disk specified as the argument to the hot spare disk pool. The storage pool identifier is the storage pool to which the physical disk belongs. If the arguments are invalid, `hotspare add` returns a non-zero exit status and relevant error message.

Removing a Physical Disk from Hot Spare Disk Pool (hotspare remove)

SYNOPSIS

```
hotspare remove pdisk_id
```

ARGUMENTS

<i>pdisk_id</i>	Physical disk identifier (UUID or serial number)
-----------------	--

RESULT

A successful `hotspare remove` command removes the physical disk specified in the argument from the hot spare disk pool. If the argument passed is invalid, `hotspare remove` returns a non-zero exit status and relevant error message.

Configuring Physical Libraries

This chapter describes the commands that you use to configure the physical library for the NearStore VTL appliance. It consists of the following sections:

- “Listing Available Physical Libraries (plib list),” page 67
- “Showing Physical Library Details (plib show),” page 68
- “Discovering a Physical Library Name or UUID (plib lookup),” page 68
- “Listing Inventory Details (plib inventory),” page 69
- “Adding a Physical Library (plib add),” page 69
- “Removing a Physical Library (plib remove),” page 70
- “Exporting a Volume (plib move_to_eep),” page 70
- “Canceling an Export Request (plib eep_cancel),” page 70
- “Polling EEP (plib eep_check),” page 71
- “Rescanning Inventory (plib rescan),” page 71

Listing Available Physical Libraries (plib list)

SYNOPSIS

```
plib list [plib_id]
```

ARGUMENTS

<i>plib_id</i>	Physical library ID to be listed
----------------	----------------------------------

RESULT

If no argument is specified, `plib_list` lists all the connected physical libraries. If `plib_id` is provided, `plib_list` prints the physical library serial number if that library is connected to the VTL. On failure, `plib_list` returns a non-zero exit status and relevant error message.

Showing Physical Library Details (plib show)

SYNOPSIS

```
plib show [plib_id]
```

ARGUMENTS

<i>plib_id</i>	ID of the physical library ID
----------------	-------------------------------

RESULT

A successful `plib show` command displays details for all the physical libraries connected to the VTL, in the following format:

```
<Name> <Address Info> <Serial No> <Device Model> <#Slots> <#Drives>  
<#Mailboxes>
```

If the `plib_id` is specified, details for only the corresponding physical library are printed. If `plib_id` is invalid, `plib show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> plib show  
  1. Physical Library:-  
  
                Name: PLIB0001  
        Address Info: 0d  
    Serial Number: d6dc8bcc747aa09803230c  
    Device Model: NETAPP    VTL  
          #Slots: 1023  
          #Drives: 1  
        #Mailboxes: 1
```

Discovering a Physical Library Name or UUID (plib lookup)

SYNOPSIS

```
plib lookup plib_id|plib_uuid
```

ARGUMENTS

<i>plib_id</i>	Name or serial number of the physical library
<i>plib_uuid</i>	UUID of the physical library

RESULT

If the argument is *plib_id*, the `plib lookup` command displays the UUID of the corresponding physical library. If the argument is *plib_uuid*, the `plib lookup` command displays the name or physical library serial number for the corresponding physical library. If the argument passed is invalid, `plib lookup` returns a non-zero exit status and relevant error message.

Listing Inventory Details (plib inventory)

SYNOPSIS

```
plib inventory plib_id [index] [count]
```

ARGUMENTS

<i>plib_id</i>	Name or serial number of the physical library
<i>index</i>	Index from which the listing should begin
<i>count</i>	Number of entries to be listed

RESULT

A successful `plib inventory` command displays inventory details for each slot, in the following format:

```
<Slot Number> <Barcode> <Data Cloned> <Data Exported> <Virtual Library>
<Date Data Written> <Export Needed>
```

If the argument passed is invalid, `plib inventory` returns a non-zero exit status and relevant error message.

Adding a Physical Library (plib add)

SYNOPSIS

```
plib add device_serialno nickname
```

ARGUMENTS

<i>device_serialno</i>	Serial number of the physical library
<i>nickname</i>	Nickname of the physical library

RESULT

A successful `plib add` command adds the physical library to the appliance configuration. If the argument passed is invalid, `plib add` returns a non-zero exit status and relevant error message.

Removing a Physical Library (plib remove)

SYNOPSIS

```
plib remove plib_id
```

ARGUMENTS

<i>plib_id</i>	Name or serial number of the physical library
----------------	---

RESULT

A successful `plib remove` command removes the physical library specified in the argument from the enabled physical library list. If the argument is invalid, `plib remove` returns a non-zero exit status and relevant error message.

Exporting a Volume (plib move_to_eep)

SYNOPSIS

```
plib move_to_eep plib_id barcode
```

ARGUMENTS

<i>plib_id</i>	Name or serial number of the physical library
<i>barcode</i>	Barcode of the physical tape to be exported

RESULT

A successful `plib move_to_eep` command exports the specified tape. On failure, `plib move_to_eep` returns a non-zero exit status and relevant error message.

Canceling an Export Request (plib eep_cancel)

SYNOPSIS

```
plib eep_cancel plib_id barcode
```

ARGUMENTS

<i>plib_id</i>	Name or serial number of the physical library
<i>barcode</i>	Barcode of the physical tape to be exported

RESULT

A successful `plib eep_cancel` command cancels export for the specified tape. On failure, `plib eep_cancel` returns a non-zero exit status and relevant error message.

Polling EEP (`plib eep_check`)

SYNOPSIS

```
plib eep_check plib_id
```

ARGUMENTS

<i>plib_id</i>	Name or serial number of the physical library
----------------	---

RESULT

A successful `plib eep_check` command injects the tape from EEP to the library. On failure, `plib eep_check` returns a non-zero exit status and relevant error message.

Rescanning Inventory (`plib rescan`)

SYNOPSIS

```
plib rescan plib_id
```

ARGUMENTS

<i>plib_id</i>	Name or serial number of the physical library
----------------	---

RESULT

A successful `plib rescan` command causes the physical tape library to rescan its inventory of tapes to show updated information. If the argument is invalid, `plib rescan` returns a non-zero exit status and relevant error message.

Configuring Physical Tape Drives

This chapter describes the commands that you use to configure the physical tape drive of the NearStore VTL appliance. It consists of the following sections:

- “Listing Physical Tape Drives (pdrive list),” page 73
- “Showing Physical Tape Drive Details (pdrive show),” page 74
- “Discovering a Physical Tape Drive Name or UUID (pdrive lookup),” page 74
- “Showing I/O Statistics for Physical Tape Drive (pdrive stats),” page 75
- “Adding a Physical Tape Drive (pdrive add),” page 75
- “Removing a Physical Tape Drive (pdrive remove),” page 75
- “Enabling a Physical Tape Drive (pdrive enable),” page 76
- “Disabling a Physical Tape Drive (pdrive disable),” page 76
- “Listing Physical Tape Drive Content (pdrive contents),” page 76

Listing Physical Tape Drives (pdrive list)

SYNOPSIS

```
pdrive list [pdrive_id]
```

ARGUMENTS

<i>pdrive_id</i>	ID or serial number of the physical tape drive
------------------	--

RESULT

A successful `pdrive list` command lists all the attached physical tape drives. On failure, `pdrive list` returns a non-zero exit status and relevant error message.

Showing I/O Statistics for Physical Tape Drive (pdrive stats)

SYNOPSIS

```
pdrive stats pdrive_id
```

ARGUMENTS

<i>pdrive_id</i>	ID or serial number of the physical tape drive
------------------	--

RESULT

A successful `pdrive stats` command displays the I/O statistics for the physical tape drive specified as the argument. If no argument is specified, statistics for all the physical tape drives are printed. If the argument is invalid, `pdrive stats` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> pdrive stats
```

```
1. Physical Tape I/O Statistics:-
```

```
      Name: TAPE0001
      Volume:
      MB/Sec: 0.000000
```

Adding a Physical Tape Drive (pdrive add)

SYNOPSIS

```
pdrive add device_serialno nickname
```

ARGUMENTS

<i>device_serialno</i>	Serial number of the physical drive to be added
<i>nickname</i>	Nickname of the physical drive to be added

RESULT

A successful `pdrive add` command adds the physical drive to the appliance configuration. If the argument is invalid, `pdrive add` returns a non-zero exit status and relevant error message.

Removing a Physical Tape Drive (pdrive remove)

SYNOPSIS

```
pdrive remove pdrive_id
```

ARGUMENTS

<code>pdrive_id</code>	ID or serial number of the physical tape drive
------------------------	--

RESULT

A successful `pdrive remove` command removes the specified physical tape drive from the appliance configuration. If the argument is invalid, `pdrive remove` returns a non-zero exit status and relevant error message.

Enabling a Physical Tape Drive (`pdrive enable`)

SYNOPSIS

```
pdrive enable pdrive_id
```

ARGUMENTS

<code>pdrive_id</code>	ID or serial number of the physical tape drive
------------------------	--

RESULT

A successful `pdrive enable` command enables the VTL to use the physical tape drive specified to export and import data. If the argument is invalid, `pdrive enable` returns a non-zero exit status and relevant error message.

Disabling a Physical Tape Drive (`pdrive disable`)

SYNOPSIS

```
pdrive disable pdrive_id
```

ARGUMENTS

<code>pdrive_id</code>	ID or serial number of the physical tape drive
------------------------	--

RESULT

A successful `pdrive disable` command disables the VTL to use the physical tape drive specified to export and import data. If the argument is invalid, `pdrive disable` returns a non-zero exit status and relevant error message.

Listing Physical Tape Drive Content (`pdrive contents`)

SYNOPSIS

```
pdrive contents
```

ARGUMENTS

None

RESULT

A successful `pdrive contents` command displays the barcode or labels of the tapes that are loaded into the physical tape drive. On failure, `pdrive contents` returns a non-zero exit status and relevant error message.

Configuring Customer Information

This chapter describes the commands that you use to configure customer information for the NearStore VTL appliance. It consists of the following sections:

- “Listing Customers (customer list),” page 79
- “Showing Customer Details (customer show),” page 80
- “Discovering a Customer Name or UUID (customer lookup),” page 80
- “Removing a Customer (customer remove),” page 81
- “Adding a New Customer (customer add),” page 81
- “Updating Customer Information (customer update),” page 82
- “Setting the Password for a Customer (customer passwd),” page 82

Listing Customers (customer list)

SYNOPSIS

```
customer list [customer_id]
```

ARGUMENTS

<i>customer_id</i>	Customer name to be listed
--------------------	----------------------------

RESULT

A successful `customer list` command displays all customers that have been added to the configuration of the NearStore VTL appliance. If *customer_id* is invalid, `customer list` returns a non-zero exit status and relevant error message.

RESULT

If *customer_id* is passed as the argument, the `customer lookup` command displays the UUID of the corresponding customer. If *customer_uuid* is passed as the argument, the `customer lookup` command displays the customer name of the corresponding customer UUID. If the argument passed is invalid, `customer lookup` returns a non-zero exit status and relevant error message.

Removing a Customer (customer remove)

SYNOPSIS

```
customer remove customer_id
```

ARGUMENTS

<i>customer_id</i>	Unique name assigned to the customer
--------------------	--------------------------------------

RESULT

A successful `customer remove` command removes the customer specified from the current appliance configuration. If *customer_id* is invalid, `customer remove` returns a non-zero exit status and relevant error message.

Adding a New Customer (customer add)

SYNOPSIS

```
customer add customer_name login_id number [contact_name] [email]
[day_phone] [night_phone] [mobile_phone] [pager]
```

ARGUMENTS

<i>customer_name</i>	Unique name to be assigned to the customer
<i>login_id</i>	User ID with which the customer will log in to the appliance
<i>number</i>	Unique tracking number of customer
<i>contact_name</i>	Contact name of the customer
<i>email</i>	E-mail address of the customer
<i>day_phone</i>]	Daytime phone number of the customer
<i>night_phone</i>	Evening phone number of the customer
<i>mobile_phone</i>	Mobile phone number of the customer
<i>pager</i>	Pager number of the customer

RESULT

If all the arguments are valid, the `customer add` command adds a new customer to the VTL configuration. When a customer is created, virtual libraries can be added and associated with that customer. If the arguments are invalid or if the command fails, `customer add` returns a non-zero exit status and relevant error message.

Updating Customer Information (customer update)

SYNOPSIS

```
customer update customer_id [name] [login_id] [number] [contact_name]  
[email] [day_phone] [night_phone] [mobile_phone] [pager]
```

ARGUMENTS

<i>customer_id</i>	Unique ID of the customer whose information is to be updated
<i>name</i>	The new name to be updated
<i>login_id</i>	User ID with which the customer will log in to the appliance
<i>number</i>	Unique tracking number of customer
<i>contact_name</i>	Contact name of the customer
<i>email</i>	E-mail address of the customer
<i>day_phone</i>	Daytime phone number of the customer
<i>night_phone</i>	Evening phone number of the customer
<i>mobile_phone</i>	Mobile phone number of the customer
<i>pager</i>	Pager number of the customer

RESULT

If the *customer_id* is valid, the `customer update` command updates the customer information with the values specified. If the arguments are invalid, `customer update` returns a non-zero exit status and relevant error message.

Setting the Password for a Customer (customer passwd)

SYNOPSIS

```
customer passwd customer_id
```

ARGUMENTS

<i>customer_id</i>	ID of customer whose password is to be set
--------------------	--

RESULT

A successful `customer passwd` command sets the customer password to the new password specified by the user. On failure, `customer passwd` returns a non-zero exit status and relevant error message.

EXAMPLE

The `customer passwd` command is interactive. It asks for the new password twice to avoid typing errors.

```
VTL> customer passwd customer_id  
Enter the Password:*****  
Re-enter the Password:*****
```


Configuring Storage Pools

This chapter describes the commands that you use to configure storage pools for the NearStore VTL appliance. It consists of the following sections:

- “Listing Names of Storage Pools (sp list),” page 85
- “Showing Storage Pool Details (sp show),” page 86
- “Discovering a Storage a Pool Name or UUID (sp lookup),” page 86
- “Adding a Storage Pool to the Appliance (sp add),” page 87
- “Removing a Storage Pool from the Appliance (sp remove),” page 87
- “Updating Storage Pool Information (sp update),” page 88

Listing Names of Storage Pools (sp list)

SYNOPSIS

```
sp list [sp_id]
```

ARGUMENTS

<i>sp_id</i>	Name of the storage pool
--------------	--------------------------

RESULT

A successful `sp list` command displays a list of all the storage pools in the configuration. If an *sp_id* is specified, the name of that storage pool is shown, if that storage pool is in the configuration. On failure, `[sp list]` returns a non-zero exit status and relevant error message.

Showing Storage Pool Details (sp show)

SYNOPSIS

`sp show [sp_id]`

ARGUMENTS

<code>sp_id</code>	Name of the storage pool
--------------------	--------------------------

RESULT

If `sp_id` is provided, the `sp show` command displays details for that storage pool, in the following format:

<Name> <Description> <Total Capacity> <Allocated Capacity> <Used Capacity> <Available Capacity>

If no argument is specified, details of all the storage pools are listed. If the argument is invalid, `sp show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> sp show
  1. Storage Pool:-
                                     Name: Default Storage Pool
                                Description: must not be deleted
                                Total Capacity: 501865250816
                                Allocated Capacity: 10737942528
                                Used Capacity: 63443968
                                Available Capacity: 501801806848
```

Discovering a Storage a Pool Name or UUID (sp lookup)

SYNOPSIS

`sp lookup sp_id|sp_uuid`

ARGUMENTS

<code>sp_id</code>	Name of the storage pool
<code>sp_uuid</code>	UUID of the storage pool

RESULT

If the argument is `sp_id`, the `sp lookup` command displays the UUID of the corresponding storage pool. If the argument is `sp_uuid`, the `sp lookup` command displays the name of the corresponding storage pool. If the argument passed is invalid, `sp lookup` returns a non-zero exit status and relevant error message.

Adding a Storage Pool to the Appliance (sp add)

SYNOPSIS

```
sp add name description
```

ARGUMENTS

<i>name</i>	Name of the storage pool being created
<i>description</i>	Description of the storage pool

RESULT

If the arguments passed are valid, `sp add` adds a storage pool to the configuration of your NearStore VTL appliance. If arguments are invalid or if `sp add` fails, the command returns a non-zero exit status and relevant error message. Before adding storage pools, you need to enable Use Storage Pools option in Advanced Options.

Any disks that you add to your NearStore VTL appliance before you enable the Use Storage Pools option are associated with the default storage pool.

Note

You cannot move these disks from the default to a newly created storage pool.

Removing a Storage Pool from the Appliance (sp remove)

SYNOPSIS

```
sp remove sp_id
```

ARGUMENTS

<i>sp_id</i>	Name of the storage pool
--------------	--------------------------

RESULT

If `sp_id` is valid, the `sp remove` deletes the storage pool specified as the argument from the configuration of your appliance. If the argument is invalid, the `sp remove` returns a non-zero exit status and relevant error message.

Note

Before you delete a storage pool, you first must delete any virtual libraries that are associated with it.

Updating Storage Pool Information (sp update)

SYNOPSIS

```
sp update sp_id [name] [description]
```

ARGUMENTS

<i>sp_id</i>	Name of the storage pool
<i>name</i>	New name of the storage pool
<i>description</i>	Text description of the storage pool

RESULT

If *sp_id* is valid, the `sp update` command updates the storage pool information with the user-specified values. If *sp_id* is invalid or if the specified storage pool is not present in the appliance, `sp update` returns a non-zero exit status and relevant error message.

CHAPTER

23

Configuring Virtual Libraries

This chapter describes the commands that you use to configure virtual libraries for the NearStore VTL appliance. It consists of the following sections:

- “Listing Virtual Libraries (vlib list),” page 89
- “Showing Virtual Library Details (vlib show),” page 90
- “Discovering a Virtual Library Name or UUID (vlib lookup),” page 91
- “Creating a Virtual Library (vlib add),” page 91
- “Removing a Virtual Library (vlib remove),” page 93
- “Updating Virtual Library Information (vlib update),” page 93
- “Importing a Volume into a Virtual Library (vlib import),” page 94
- “Exporting a Volume from a Virtual Library (vlib export),” page 95
- “Listing Inventory Details of a Virtual Library (vlib inventory),” page 95

Listing Virtual Libraries (vlib list)

SYNOPSIS

```
vlib list [vlib_id]
```

ARGUMENTS

<i>vlib_id</i>	Virtual library name
----------------	----------------------

RESULT

A successful `vlib list` command displays names of all the virtual libraries in the appliance. If the *vlib_id* is specified, the name of that virtual library is printed if the library is present. If the *vlib_id* is invalid or if `vlib list` fails, the command returns a non-zero exit status and relevant error message.

Showing Virtual Library Details (vlib show)

SYNOPSIS

```
vlib show [vlib_id]
```

ARGUMENTS

<code>vlib_id</code>	Name of virtual library
----------------------	-------------------------

RESULT

If `vlib_id` is valid, the `vlib show` command displays details for the virtual library, in the following format:

```
<Name> <Serial Number> <Customer> <Vendor ID> <Product ID> <Tape Drive Type>
<Allocated Capacity> <Used Capacity> <Data Written> <#Volumes>
<#Slots> <#Drives> <Compression> <Tape Cloning Enabled>
<Auto move Tape from EEP to Slot> <Storage Pool>
```

If no argument is passed, `vlib show` displays details for all the virtual libraries present in the appliance. If `vlib_id` is invalid, `vlib show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> vlib show
```

```
1. Virtual Library:-
```

```

                                Name: vlib01
                        Serial Number: de821798b1b5145e958054
                        Customer: admin
                        Vendor ID: NETAPP
                        Product ID: VTL
                        Tape Drive Type: SDLT
Allocated Capacity: 0
Used Capacity: 0
Data Written: 0
#Volumes: 20
#Slots: 20
#Drives: 2
#Mailboxes: 1
#Transports: 1
First Slot: 0
First Drive: 20
First Mailbox: 22
First Transport: 23
Tape Smart Sizing: Disabled
Shadow Tape Enabled: False
Tape Cloning Enabled: False
Auto move Tape from EEP to Slot: True
```

```
Storage Pool: Default Storage Pool
Host Access Scope: Dedicated
Port Access Scope: Single-Port
Disk Allocation per Tape: 0.000000
Disk Compression: Disabled
```

Discovering a Virtual Library Name or UUID (vlib lookup)

SYNOPSIS

```
vlib lookup vlib_id|vlib_serial_number|vlib_uuid
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library
<i>vlib_serial_number</i>	Serial number of the virtual library
<i>vlib_uuid</i>	UUID of the virtual library

RESULT

If the argument is *vlib_id*, the `vlib lookup` command displays the UUID of the corresponding virtual library. If the argument is *vlib_serial_number*, the `vlib lookup` command displays the UUID of the corresponding virtual library. If the argument is *vlib_uuid*, the `vlib lookup` command displays the name of the corresponding virtual library. If the argument passed is invalid, `vlib lookup` returns a non-zero exit status and relevant error message.

Creating a Virtual Library (vlib add)

SYNOPSIS

```
vlib add name lib_type vtdt_id slots drives transports [customer_id]  
[sp_id] [port_id] [host_id] [vendor_id] [product_id] [auto_move_to_slot]  
[enable_cloning] [enable_disk_compression] [enable_tape_smart_sizing]  
[fully_load_with_vtapes] [start_vtape_name] [disk_allocation_per_tape]  
[auto_assign_first_values] [first_slot] [first_drive] [first_mailbox]  
[first_transport] [enable_shadow_tape] [shadow_tape_retention_time]
```

ARGUMENTS

<i>name</i>	Name of the virtual library that you want to create.
<i>lib_type</i>	Library type that you want to create.
<i>vtdt_id</i>	Virtual tape drive type.

<i>slots</i>	Slot count of the virtual library.
<i>drives</i>	Drive count of the virtual library.
<i>transports</i>	Transport count of the virtual library.
<i>customer_id</i>	Customer name (if use customer is enabled).
<i>sp_id</i>	Storage pool name (if Use Storage Pool is enabled).
<i>port_id</i>	Port on which the library will be visible.
<i>host_id</i>	Host to which the virtual library access is to be given.
<i>vendor_id</i>	Vendor ID if tape drive type is Custom.
<i>product_id</i>	Product ID if tape drive type is Custom.
<i>auto_move_to_slot</i>	NearStore VTL automatically moves a created or imported virtual tape from the entry port to an available slot. If this is <i>not</i> set, the backup application moves it.
<i>enable_cloning</i>	NearStore VTL creates a physical copy of the virtual tape each time that the virtual tape that has been modified is removed from the virtual drive.
<i>enable_disk_compression</i>	Enables disk compression for tapes.
<i>enable_tape_smart_sizing</i>	Set if you want compression enabled.
<i>fully_load_with_vtapes</i>	Set this if you want all slots to be loaded with tapes.
<i>start_vtape_name</i>	Name of the start VTV if Fully Load is set.
<i>disk_allocation_per_tape</i>	The amount of disk space that NearStore VTL should allocate per virtual tape.
<i>auto_assign_first_values</i>	If set, VTL assigns values to the first slot, the first drive, the first mailbox, and the first transport.
<i>first_slot</i>	The value for the first slot if <i>auto_assign_first_values</i> is <i>not</i> set.
<i>first_drive</i>	The value for the first drive if <i>auto_assign_first_values</i> is <i>not</i> set.
<i>first_mailbox</i>	The value for the first mailbox if <i>auto_assign_first_values</i> is <i>not</i> set.
<i>first_transport</i>	The value for the first transport if <i>auto_assign_first_values</i> is <i>not</i> set.
<i>enable_shadow_tape</i>	If Use Shadow Tape is enabled in Advanced Options, you can enable it at the virtual library level.
<i>shadow_tape_retention_time</i>	The value for retention time for deleted VTVs if Use Shadow Pool is enabled.

RESULT

If all the arguments are valid, the `vlib add` command creates a new virtual library with user-specified values. If any of the arguments is invalid, `vlib add` returns a non-zero exit status and relevant error message.

Removing a Virtual Library (`vlib remove`)

SYNOPSIS

```
vlib remove
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library to be removed
----------------	---

RESULT

If the specified virtual library is present, the `vlib remove` command deletes it. By default, all the virtual tapes in the virtual library are exported. You must delete the virtual tapes before deleting the virtual library. If the command fails or if *vlib_id* is invalid, `vlib remove` returns a non-zero exit status and relevant error message.

Updating Virtual Library Information (`vlib update`)

SYNOPSIS

```
vlib update vlib_id [name] [slots] [drives] [auto_move_to_slot]
[enable_cloning] [enable_disk_compression] [enable_shadow_tape]
[shadow_tape_retention_time] [disk_allocation_per_tape]
[auto_assign_first_values] [first_slot] [first_drive]
[first_mailbox] [first_transport] [enable_tape_smart_sizing]
[tape_smart_sizing_type]
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library to be removed.
<i>name</i>	Name to be updated.
<i>slots</i>	Slot count of virtual library.
<i>drives</i>	Drive count of virtual library.
<i>auto_move_to_slot</i>	NearStore VTL automatically moves a created or imported virtual tapes from the entry port to an available slot. If this is not set, the backup application moves them.

<code>enable_cloning</code>	NearStore VTL creates a physical copy of the virtual tape each time that the virtual tape that has been modified is removed from the virtual drive.
<code>enable_disk_compression</code>	Enables disk compression for tapes.
<code>enable_shadow_tape</code>	If Use Shadow Tape is enabled in Advanced Options, you can enable it at virtual library level.
<code>shadow_tape_retention_time</code>	The value for retention time to be used for deleted VTVs if Enable Shadow Pool is enabled.
<code>disk_allocation_per_tape</code>	How much disk space NearStore VTL should allocate per virtual tape.
<code>auto_assign_first_values</code>	If set, VTL assigns values to the first slot, the first drive, the first mailbox, and the first transport.
<code>first_slot</code>	The value for first slot if <code>auto_assign_first_values</code> is <i>not</i> set.
<code>first_drive</code>	The value for first drive if <code>auto_assign_first_values</code> is <i>not</i> set.
<code>first_mailbox</code>	The value for first mailbox if <code>auto_assign_first_values</code> is <i>not</i> set.
<code>first_transport</code>	The value for first transport if <code>auto_assign_first_values</code> is <i>not</i> set.
<code>enable_tape_smart_sizing</code>	Set if you want compression enabled.
<code>tape_smart_sizing_type</code>	If Tape Smart Sizing is enabled, fill in which type of compression to use. The possible values are: LZS, <code>lzs_c</code> (LZS Conservative), ALDC, <code>aldc_c</code> (ALDC Conservative), <code>ibm_lto</code> , and none.

RESULT

If `vlib_id` is valid, the `vlib update` command updates the corresponding virtual library information with user-specified values. If `vlib_id` is invalid or if `vlib update` fails to update information, the command returns a non-zero exit status and relevant error message.

Importing a Volume into a Virtual Library (vlib import)

SYNOPSIS

```
vlib import vlib_id barcode [quick_load] [read_only] [found_only]
[cache_only]
```

ARGUMENTS

<i>vlib_id</i>	Name, UUID, or serial number of the virtual tape library into which to import the tape.
<i>barcode</i>	Barcode of the tape to be imported.
<i>quick_load</i>	Loads only the first few bytes of data
<i>read_only</i>	Physical tape is imported as read only.
<i>found_only</i>	Imports the tape if it is present in VTL configuration data (physical lib inventory or shadow tape pool).
<i>cache_only</i>	Imports the tape if it is present in shadow tape only.

RESULT

A successful `vlib import` command has an exit status of 0 and imports a volume. A failed `vlib import` command returns a non-zero exit status and relevant error message.

Exporting a Volume from a Virtual Library (vlib export)

SYNOPSIS

```
vlib export vtape_id
```

ARGUMENTS

<i>vtape_id</i>	Barcode or name of the volume to be exported
-----------------	--

RESULT

If *vtape_id* is valid, the `vlib export` command creates an export request for that VTL. When a tape is exported, if Shadow Tape Pool is enabled, the tape also is moved to the shadow tape pool. If *vtape_id* is invalid, `vlib export` returns a non-zero exit status and relevant error message.

Listing Inventory Details of a Virtual Library (vlib inventory)

SYNOPSIS

```
vlib inventory vlib_id [index] [count]
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library
<i>index</i>	Index from which the listing should begin
<i>count</i>	Number of entries to be listed

RESULT

If the `vlib_id` is valid, the `vlib inventory` command prints details for that virtual library. If the argument is invalid, `vlib inventory` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> vlib inventory vlib01 0 1
```

```
1. Virtual Library Inventory:-
```

```
Slot Number: 0
```

```
Barcode: vlib1_vol01
```

Configuring Virtual Tapes

This chapter describes the commands that you use to configure virtual tapes for the NearStore VTL appliance. It consists of the following sections:

- “Listing Virtual Tapes (vtape list),” page 97
- “Showing Virtual Tape Details (vtape show),” page 98
- “Discovering a Virtual Tape Name or UUID (vtape lookup),” page 99
- “Creating a Virtual Tape (vtape add),” page 99
- “Removing a Virtual Tape (vtape remove),” page 100
- “Updating a Virtual Tape (vtape update),” page 100
- “Moving a Volume from One Virtual Library to Another (vtape move),” page 101
- “Loading a Volume into a Virtual Library Drive (vtape load),” page 101
- “Unloading a Volume from a Virtual Library Drive (vtape unload),” page 101

Listing Virtual Tapes (vtape list)

SYNOPSIS

```
vtape list [vtape_id] [index] [count]
```

ARGUMENTS

<i>vtape_id</i>	Barcode or name of the virtual tape
<i>index</i>	Index from which the listing should begin
<i>count</i>	Number of entries to be listed

RESULT

A successful `vtape list` command displays a list of all the virtual tapes in the appliance. If the `vtape_id` is specified, `vtape list` prints the name of the virtual tape if it is present in the appliance. If `vtape list` fails or if `vtape_id` is invalid, the command returns a non-zero exit status and relevant error message.

Showing Virtual Tape Details (`vtape show`)

SYNOPSIS

```
vtape show [vtape_id] [index] [count]
```

ARGUMENTS

<i>vtape_id</i>	Barcode or name of the virtual tape
<i>index</i>	Index from which the listing should begin
<i>count</i>	Number of entries to be listed

RESULT

If `vtape_id` is provided, the `vtape show` command displays details for that vtape, in the following format:

```
<Name> <Customer> <Library> <#Files> <Read Only> <Export Needed>  
<Raw Capacity> <Data Written> <Remaining Space> <Disk Usage>  
<Disk Compression> <Tape Smart Sizing>
```

If no argument is provided, the `vtape show` command displays details of all the tapes present in the appliance. If `vtape_id` is invalid, `vtape show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> vtape show index=0 count=1
```

```
1. Virtual Tape:-
```

```
                Name: vlib1_vol01  
            Customer: admin  
            Library: vlib01  
              #Files: 0  
        Read Only: False  
    Export Needed: False  
    Raw Capacity: 160000000000  
    Data Written: 0  
Remaining Space: 160000000000  
    Disk Usage: 0  
Disk Compression: Enabled  
Tape Smart Sizing: Disabled
```

Discovering a Virtual Tape Name or UUID (vtape lookup)

SYNOPSIS

```
vtape lookup vtape_id|vtape_uuid
```

ARGUMENTS

<i>vtape_id</i>	Name of the virtual tape
<i>vtape_uuid</i>	UUID of the virtual tape

RESULT

If the argument is *vtape_id*, the `vtape lookup` command displays the UUID of the corresponding virtual tape. If the argument is *vtape_uuid*, the `vtape lookup` command displays the virtual tape name of the corresponding virtual tape UUID. If the argument passed is invalid, `vtape lookup` returns a non-zero exit status and relevant error message.

Creating a Virtual Tape (vtape add)

SYNOPSIS

```
vtape add vlib_id start_name count
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library to which a virtual tape is to be added.
<i>start_name</i>	Name of the first tape in the sequence out of <i>n</i> tapes to be added.
<i>count</i>	Number of tapes to be added. When you create multiple virtual tapes, the NearStore VTL increments their labels based on the starting tape label.

RESULT

If the *vlib_id* is valid, the `vtape add` command creates *count* number of virtual tapes. By default, the virtual tapes that you create take on the characteristics of the associated virtual library. This includes the capacity of the virtual tape and whether it uses Tape Smart Sizing.

Note

Because there is a one-to-one correlation between virtual tapes and physical tapes, it is important that the labels (virtual barcodes) of your virtual tapes match (including case) the barcodes of the corresponding physical tapes.

Removing a Virtual Tape (vtape remove)

SYNOPSIS

```
vtape remove vlib_id
```

ARGUMENTS

<i>vlib_id</i>	Name or barcode of the virtual tape to be removed
----------------	---

RESULT

If *vlib_id* is valid, the `vtape remove` command deletes the virtual tape from the appliance. When a tape is deleted, all the data in the tape is also deleted. If the *vlib_id* is invalid, `vtape remove` returns a non-zero exit status and relevant error message.

Updating a Virtual Tape (vtape update)

SYNOPSIS

```
vtape update vlib_id [export_needed] [read_only]
[enable_disk_compression] [enable_tape_smart_sizing]
[tape_smart_sizing_type]
```

ARGUMENTS

<i>vlib_id</i>	Name or barcode of the virtual tape to be updated.
<i>export_needed</i>	By setting this flag, you can force the export of the tape.
<i>read_only</i>	By setting this flag, you can make the tape read-only.
<i>enable_disk_compression</i>	Enables disk compression for tapes.
<i>enable_tape_smart_sizing</i>	Set if you want compression enabled.
<i>tape_smart_sizing_type</i>	If Tape Smart Sizing is enabled, specify which type of compression to use. The possible values are: LZS, <i>lzs_c</i> (LZS Conservative), ALDC, <i>aldc_c</i> (ALDC Conservative), <i>ibm_lto</i> , and none.

RESULT

If the *vlib_id* is valid, the `vtape update` command updates the virtual tape with the user-specified values. If the *vlib_id* is invalid, `vtape update` returns a non-zero exit status and relevant error message.

Note

A virtual tape that has been imported from physical tape does not need to be copied to a physical tape unless it has been modified.

Moving a Volume from One Virtual Library to Another (vtape move)

SYNOPSIS

```
vtape move vtape_id dest_lib_id
```

ARGUMENTS

<i>vtape_id</i>	Barcode or name of the tape volume to be moved
<i>dest_lib_id</i>	Name of the virtual library to which the volume is to be moved

RESULT

If the arguments passed are valid, the `vtape move` command moves the tape volume from the source library to the destination library. If `vtape move` fails or if the arguments are invalid, the command returns a non-zero exit status and relevant error message.

Loading a Volume into a Virtual Library Drive (vtape load)

SYNOPSIS

```
vtape load vtape_id drive
```

ARGUMENTS

<i>vtape_id</i>	Barcode, name, or UUID of the virtual tape
<i>drive</i>	Drive ID in which the volume is to be loaded (The argument <code>drive_id=0</code> implies the first drive in the library, and so on.)

RESULT

If the arguments are valid, the `vtape load` command loads the volume into the drive for the virtual library to which the volume belongs. If arguments are invalid, or if the load fails, `vtape load` returns a non-zero exit status and relevant error message.

Unloading a Volume from a Virtual Library Drive (vtape unload)

SYNOPSIS

```
vtape unload vtape_id [slot]
```

ARGUMENTS

<i>vtape_id</i>	Barcode, name, or UUID of the virtual tape
<i>slot</i>	Slot to which the tape is placed after being unload

RESULT

If the arguments are valid, the `vtape unload` command unloads the tape volume from the drive to the slot specified in the argument in the corresponding virtual library. If the arguments are invalid or if the unload fails, `vtape unload` returns a non-zero exit status and relevant error message.

Configuring Virtual Tape Drive Type

This chapter describes the commands that you use to configure the virtual tape drive type for the NearStore VTL appliance. It consists of the following sections:

- “Listing a Virtual Tape Drive Type (vtdt list),” page 103
- “Showing Virtual Tape Drive Type Details (vtdt show),” page 104
- “Discovering a Virtual Tape Drive Type Name or UUID (vtdt lookup),” page 104
- “Adding a New Virtual Tape Drive Type (vtdt add),” page 105
- “Removing a Virtual Tape Drive Type (vtdt remove),” page 105
- “Updating a Virtual Tape Drive Type (vtdt update),” page 106

Listing a Virtual Tape Drive Type (vtdt list)

SYNOPSIS

```
vtdt list [vtdt_id]
```

ARGUMENTS

<i>vtdt_id</i>	Virtual tape drive type name
----------------	------------------------------

RESULT

A successful `vtdt list` command lists all the virtual tape drive types available in the appliance configuration. If the *vtdt_id* is specified, `vtdt list` lists the name if that virtual tape drive type is present in the appliance. If the *vtdt_id* is invalid, `vtdt list` returns a non-zero exit status and relevant error message.

Showing Virtual Tape Drive Type Details (vtdt show)

SYNOPSIS

`vtdt show [vtdt_id]`

ARGUMENTS

<code>vtdt_id</code>	Virtual tape drive type name
----------------------	------------------------------

RESULT

If the `vtdt_id` is specified, `vtdt show` displays details for that virtual tape drive type, in the following format:

<Name> <Raw Capacity> <Disk Allocation per Tape> <Vendor ID>
<Product ID> <Tape Smart Sizing>

If no argument is specified, details are printed for all the virtual tape drive types. If the argument is invalid, `vtdt show` returns a non-zero exit status and relevant error message.

EXAMPLE

VTL> vtdt show

1. Virtual Tape Drive Template:-

Name: AIT 1
Raw Capacity: 35.000000
Disk Allocation per Tape: 70.000000
Vendor ID: SONY
Product ID: SDX-400C
Tape Smart Sizing: ALDC

Discovering a Virtual Tape Drive Type Name or UUID (vtdt lookup)

SYNOPSIS

`vtdt lookup vtdt_id|vtdt_uuid`

ARGUMENTS

<code>vtdt_id</code>	Virtual tape drive type name
<code>vtdt_uuid</code>	UUID of the tape drive type

RESULT

If the argument is `vtdt_id`, the `vtdt lookup` command displays the UUID of the corresponding virtual tape drive type. If the argument is `vtdt_uuid`, the `vtdt lookup` command displays the name

of the corresponding virtual tape drive type. If the argument passed is invalid, `vtdt lookup` returns a non-zero exit status and relevant error message.

Adding a New Virtual Tape Drive Type (`vtdt add`)

SYNOPSIS

```
vtdt add name capacity vendor_id product_id compression_type  
disk_allocation
```

ARGUMENTS

<i>name</i>	Name of the new virtual type drive type.
<i>capacity</i>	Capacity of the virtual tape drive type.
<i>vendor_id</i>	Vendor identification by which this type of virtual tape drive identifies itself on the SAN.
<i>product_id</i>	Product identification by which this type of virtual tape drive identifies itself on the SAN.
<i>compression_type</i>	Type of compression that your physical tape drives are using. The possible values are: LZS, lzs_c(LZS Conservative), ALDC, aldc_c(ALDC Conservative), ibm_lto, none.
<i>disk_allocation</i>	Disk space that NearStore VTL should allocate per virtual tape (in GB).

RESULT

If all the arguments are valid, `vtdt add` adds a new virtual tape drive type to the appliance configuration. If any of the arguments is invalid, `vtdt add` returns a non-zero exit status and relevant error message.

Note

If a virtual tape drive type is currently being used by a virtual library, you cannot use it as the basis for a new virtual tape drive type.

Removing a Virtual Tape Drive Type (`vtdt remove`)

SYNOPSIS

```
vtdt remove vtdt_id
```

ARGUMENTS

<i>vtdt_id</i>	Virtual tape drive type name
----------------	------------------------------

RESULT

A successful `vtddt remove` command removes the virtual tape drive type specified from the appliance configuration. You cannot remove the virtual tape drive type if it is in use by a virtual library. If the argument is invalid, `vtddt remove` returns a non-zero exit status and relevant error message.

Updating a Virtual Tape Drive Type (vtddt update)

SYNOPSIS

```
vtddt update vtddt_id [name] [capacity] [vendor_id] [product_id]
[compression_type] [disk_allocation]
```

ARGUMENTS

<i>vtddt_id</i>	Name of a virtual tape drive type.
<i>name</i>	New name of the virtual tape drive type.
<i>capacity</i>	Capacity of the virtual tape drive type.
<i>vendor_id</i>	Vendor identification by which this type of virtual tape drive identifies itself on the SAN.
<i>product_id</i>	Product identification by which this type of virtual tape drive identifies itself on the SAN.
<i>compression_type</i>	Type of compression that your physical tape drives are using. The possible values are: LZS, lzs_c(LZS Conservative), ALDC, aldc_c(ALDC Conservative), ibm_lto, and none.
<i>disk_allocation</i>	Disk space that NearStore VTL should allocate per virtual tape (in GBs).

RESULT

If *vtddt_id* is valid, the `vtddt update` command updates the virtual tape type with the user-specified values. If *vtddt_id* is invalid, `vtddt update` returns a non-zero exit status and relevant error message.

Configuring Hosts

This chapter describes the commands that you use to configure hosts for the NearStore VTL appliance. It consists of the following sections:

- “Listing Host Names (host list),” page 107
- “Showing Host Details (host show),” page 108
- “Discovering a Host Name or UUID (host lookup),” page 108
- “Adding a New Host (host add),” page 109
- “Removing a Host (host remove),” page 109
- “Updating a Host (host update),” page 109

Listing Host Names (host list)

SYNOPSIS

```
host list host_id
```

ARGUMENTS

<i>host_id</i>	Name assigned to host
----------------	-----------------------

RESULT

A successful `host list` command displays the list of host names that are present in the appliance configuration. If the `host_id` is passed, `host list` prints the name of that host if that host is present in the appliance configuration. If `host list` fails, it returns a non-zero exit status and relevant error message.

Showing Host Details (host show)

SYNOPSIS

`host show host_id`

ARGUMENTS

<i>host_id</i>	Name assigned to host
----------------	-----------------------

RESULT

If the *host_id* is specified, the `host show` command displays details for that host, in the following format:

<Name> <Customer> <WWN>

If no argument is entered, `host show` displays details for all of the hosts present in the appliance. If the argument is invalid, `host show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> host show
1. Host:-
      Name: host1
      Customer: admin
      WWN: 1234123412341234
```

Discovering a Host Name or UUID (host lookup)

SYNOPSIS

`host lookup host_name|host_uuid`

ARGUMENTS

<i>host_name</i>	Name assigned to the host
<i>host_uuid</i>	UUID of the host

RESULT

If the argument is *host_name*, the `host lookup` command displays the UUID of the corresponding host. If the argument is *host_uuid*, the `host lookup` command displays the name of the corresponding host. If the argument passed is invalid, `host lookup` returns a non-zero exit status and relevant error message.

Adding a New Host (host add)

SYNOPSIS

```
host add name wwpn [customer_id]
```

ARGUMENTS

<i>name</i>	Name to be assigned to the host.
<i>wwn</i>	The world wide port name (WWN) for the host. You must enter this as a 16-character hexadecimal value.
<i>customer_id</i>	Name of the customer who owns the listed hosts.

RESULT

If the arguments are valid, the `host add` command adds the host, with specified values, to the appliance configuration. If the arguments are invalid, `host add` returns a non-zero exit status and relevant error message.

Removing a Host (host remove)

SYNOPSIS

```
host remove host_id
```

ARGUMENTS

<i>host_id</i>	Name assigned to the host
----------------	---------------------------

RESULT

A successful `host remove` command removes the host specified in the argument from the appliance configuration. When a host is deleted, all of the access mappings to that host also are deleted. If the argument is invalid, `host remove` returns a non-zero exit status and relevant error message.

Updating a Host (host update)

SYNOPSIS

```
host update host_id [name] [wwn]
```

ARGUMENTS

<i>host_id</i>	Name assigned to the host.
<i>name</i>	Name to be assigned to the host.
<i>wwn</i>	The World Wide Port Name (WWN) for the host. You must enter this as a 16-character hexadecimal value.

RESULT

A successful `host update` command updates the given host with user-specified values. If the arguments are invalid, `host update` returns a non-zero exit status and relevant error message.

Configuring Ports

This chapter describes the commands that you use to configure ports for the NearStore VTL appliance. It consists of the following sections:

- “Listing Ports (port list),” page 111
- “Showing Port Details (port show),” page 111
- “Discovering a Port Name or UUID (port lookup),” page 112

Listing Ports (port list)

SYNOPSIS

```
port list [port_id]
```

ARGUMENTS

<i>port_id</i>	Name of the port
----------------	------------------

RESULT

A successful `port list` command displays the list of front-end physical ports of the NearStore VTL appliance. If the *port_id* is specified, `port list` prints the port name if that port is present in the appliance configuration. If `port list` fails, it returns a non-zero exit status and relevant error message.

Showing Port Details (port show)

SYNOPSIS

```
port show [port_id]
```

ARGUMENTS

<i>port_id</i>	Name of the port
----------------	------------------

RESULT

If the *port_id* is specified, the `port show` command displays details of the port. If no argument is entered, details are printed for all the ports available

EXAMPLE

VTL> **port show**
1. Port:-

Name: Port1
Role: Backend
Port WWNN: 200000e08b1f4697
Port WWPN: 210100e08b3f4397
Topology: Private Loop
Port ID: 0xef
Loop ID: 0x0

Discovering a Port Name or UUID (port lookup)

SYNOPSIS

`port lookup port_name|port_uuid`

ARGUMENTS

<i>port_name</i>	Name of the port
<i>port_uuid</i>	UUID of the port

RESULT

If the argument is *port_name*, the `port lookup` command displays the UUID of the corresponding port. If the argument is *port_uuid*, the `port lookup` command displays the name of the corresponding port. If the argument passed is invalid, `port lookup` returns a non-zero exit status and relevant error message.

Showing Appliance Actions

This chapter describes the command that you use to show action details for the NearStore VTL appliance.

Showing Action Details (action show)

SYNOPSIS

```
action show [index] [count]
```

ARGUMENTS

<i>index</i>	Index from which the listing should begin
<i>count</i>	Number of entries to be listed

RESULT

A successful `action show` command displays details of all the actions currently logged in the appliance, in the following format:

```
<Code> <Description> <Priority> <Age> <Acknowledged>
```

On failure, `action show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> action show
```

```
1. Action:-
```

```

                                Code: 2970
                        Description: Hot spare disk pool needs disks \
of 250.93 GB sizes.
                                Priority: MEDIUM
                                Age: 02:30:17
                        Acknowledged: False
```


Importing or Exporting Requests

This chapter describes the commands that you use to make import or export requests for the NearStore VTL appliance. It consists of the following sections:

- “Showing Import or Export Request Details (`ireq show`),” page 115
- “Requesting the Loading of a Volume (`ireq load`),” page 116
- “Canceling a Request (`ireq cancel`),” page 116
- “Requeuing a Request (`ireq requeue`),” page 116

Showing Import or Export Request Details (`ireq show`)

SYNOPSIS

```
ireq show [index] [count]
```

ARGUMENTS

<i>index</i>	Index from which the listing should begin
<i>count</i>	Number of entries to be listed

RESULT

A successful `ireq show` command displays details of import or export requests, in the following format:

```
<Requester> <Volume> <Library> <Mode> <Direction> <Age> <State>
```

On failure, `ireq show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> ireq show
```

```
1. Mount Request:-
```

```
Requester: Application
```

```
Volume: vlib1_vol10
Library: vlib1
Mode: export tape
Direction: export
Age: 00:00:03
State: tape not found
```

Requesting the Loading of a Volume (iereq load)

SYNOPSIS

```
iereq load barcode [ptape_id]
```

ARGUMENTS

<i>barcode</i>	Barcode of the tape to be loaded
<i>ptape_id</i>	Physical tape drive from which the barcode is to be loaded

RESULT

A successful `iereq load` command loads the tape (barcode) into the recommended tape drive provided that there is a mount request for that tape in the import or export list. If *ptape_id* is specified, the VTL loads the tape from the physical tape drive corresponding to the *ptape_id*. On failure, `iereq load` returns a non-zero exit status and relevant error message.

Canceling a Request (iereq cancel)

SYNOPSIS

```
iereq cancel barcode
```

ARGUMENTS

<i>barcode</i>	Barcode of the tape to be loaded
----------------	----------------------------------

RESULT

A successful `iereq cancel` command cancels the import or export request for the tape (barcode). On failure, `iereq cancel` returns a non-zero exit status and relevant error message.

Requeuing a Request (iereq requeue)

SYNOPSIS

```
iereq requeue barcode
```

ARGUMENTS

<i>barcode</i>	Barcode of the tape to be loaded
----------------	----------------------------------

RESULT

A successful `iereq requeue` command requeues the mount request for the tape provided that the mount request for that tape has failed. On failure, `iereq requeue` returns a non-zero exit status and relevant error message.

Using the Pick List

This chapter describes the commands that you use to work with pick lists for the NearStore VTL appliance. It consists of the following sections:

- “Showing the Pick List (picklist show),” page 119
- “Notifying That a Volume Has Been Found (picklist found),” page 119
- “Notifying That a Volume Is Missing (picklist missing),” page 120

Showing the Pick List (picklist show)

SYNOPSIS

```
picklist show [index] [count]
```

ARGUMENTS

<i>index</i>	Index from which the listing should begin
<i>count</i>	Number of entries to be listed

RESULT

A successful `picklist show` command displays details of all the physical tapes that need to be mounted to physical tape drives, in the following format:

```
<Tape> <Requester> <Age>
```

On failure, `picklist show` returns a non-zero exit status and relevant error message.

Notifying That a Volume Has Been Found (picklist found)

SYNOPSIS

`picklist found barcode`

ARGUMENTS

<i>barcode</i>	Barcode of the volume found
----------------	-----------------------------

RESULT

If *barcode* is valid, the `picklist found` command notifies the appliance that the physical tape volume is found and that required import or export should be continued. On failure, `picklist found` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> picklist show
1. PickList Request:-
                                Requester: Application
                                Barcode: vlib1_vol07
                                Creation Time: Tue Nov 21 17:59:54 2006
```

Notifying That a Volume Is Missing (picklist missing)

SYNOPSIS

`picklist missing barcode`

ARGUMENTS

<i>barcode</i>	Barcode of the missing volume
----------------	-------------------------------

RESULT

If the *barcode* is valid, the `picklist missing` command informs the appliance that the physical tape volume specified is missing. On failure, `picklist missing` returns a non-zero exit status and relevant error message.

Configuring Host or Port Access

This chapter describes the commands that you use to configure host or port access for the NearStore VTL appliance. It consists of the following sections:

- “Listing Host Access (access host_list),” page 121
- “Defining Scope for Host Access (access host_scope),” page 122
- “Adding Host Access (access host_add),” page 122
- “Removing Host Access (access host_remove),” page 122
- “Listing Port Access (access port_list),” page 123
- “Defining Scope for Port Access (access port_scope),” page 123
- “Adding Port Access (access port_add),” page 124
- “Removing Port Access (access port_remove),” page 124

Listing Host Access (access host_list)

SYNOPSIS

```
access host_list
```

ARGUMENTS

None

RESULT

A successful `access host_list` command displays the host access information for the libraries, in the following format:

```
<Library> <Customer> <Device> <Host> <LUN>
```

On failure, `access host_list` returns a non-zero exit status and relevant error message.

Defining Scope for Host Access (access host_scope)

SYNOPSIS

```
access host_scope vlib_id scope [host_id]
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library
<i>scope</i>	dedicated if the host has dedicated access to the devices; shared if the host has shared access to multiple devices
<i>host_id</i>	Name of the host

RESULT

A successful `access host_scope` command sets the scope for the host access to a library or to a set of library devices. Access is set to `dedicated` when all devices in a library can be accessed by the same host. Otherwise, access is set to `shared`.

Adding Host Access (access host_add)

SYNOPSIS

```
access host_add vlib_id host_id device_id
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library
<i>host_id</i>	Name of the host
<i>device_id</i>	Robot /(Drive0,Drive1 ...)

RESULT

A successful `access host_add` command adds the host access for the device (*device_id*) in a virtual library (*vlib_id*) to the host (*host_id*). On failure, `access host_add` returns a non-zero exit status and relevant error message.

Removing Host Access (access host_remove)

SYNOPSIS

```
access host_remove vlib_id host_id device_id
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library
<i>host_id</i>	Name of the host
<i>device_id</i>	Robot /(Drive0,Drive1 ...)

RESULT

A successful `access host_remove` command removes the host access for the device (*device_id*) in a virtual library (*vlib_id*) from the host (*host_id*). On failure, `access host_remove` returns a non-zero exit status and relevant error message.

Listing Port Access (`access port_list`)

SYNOPSIS

```
access port_list [vlib_id]
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library
----------------	-----------------------------

RESULT

A successful `access port_list` command prints the port access for the library devices in the following format:

```
<Library> <Customer> <Device> <Port> <LUN>
```

On failure, `access port_list` returns a non-zero exit status and relevant error message.

Defining Scope for Port Access (`access port_scope`)

SYNOPSIS

```
access port_scope vlib_id scope [port_id]
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library
<i>scope</i>	Single_port if the device is visible on a single port; multi_port if the device is visible on multiple ports
<i>port_id</i>	Name of the port

RESULT

A successful `access port_scope` command defines the scope for the port (*port_id*) for the library (*vlib_id*) as single_port or multi_port (*scope*). On failure, `access port_scope` returns a non-zero exit status and relevant error message.

Adding Port Access (access port_add)

SYNOPSIS

```
access port_add  vlib_id  port_id  device_id
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library
<i>port_id</i>	Name of the port
<i>device_id</i>	Robot /(Drive0,Drive1 ...)

RESULT

A successful `access port_add` command adds the port access for the device (*device_id*) in a virtual library (*vlib_id*) to the port (*port_id*). On failure, `access port_add` returns a non-zero exit status and relevant error message.

Removing Port Access (access port_remove)

SYNOPSIS

```
access port_remove  vlib_id  port_id  device_id
```

ARGUMENTS

<i>vlib_id</i>	Name of the virtual library
<i>port_id</i>	Name of the port
<i>device_id</i>	Robot /(Drive0,Drive1 ...)

RESULT

A successful `access port_remove` command remove the port access for the device (*device_id*) in a virtual library (*vlib_id*) from the port (*port_id*). On failure, `access port_remove` returns a non-zero exit status and relevant error message.

Configuring Virtual Tape Drive

This chapter describes the commands that you use to configure virtual tape drives for the NearStore VTL appliance. It consists of the following sections:

- “Listing Virtual Tape Drives (vdrive list),” page 125
- “Showing Virtual Tape Drive Details (vdrive show),” page 125
- “Showing I/O Statistics for a Virtual Tape Drive (vdrive stats),” page 126

Listing Virtual Tape Drives (vdrive list)

SYNOPSIS

```
vdrive list
```

ARGUMENTS

None

RESULT

A successful `vdrive list` command lists the names of all the virtual tape drives present in the appliance configuration.

Showing Virtual Tape Drive Details (vdrive show)

SYNOPSIS

```
vdrive show
```

ARGUMENTS

None

RESULT

A successful `vdrive show` command displays details of the virtual tape drives. On failure, `vdrive show` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> vdrive show
```

```
1. Virtual Drive:-
```

```
Virtual Library: test
Virtual Drive: Drive0
Serial Number: d6deb208747aa09803230c
Vendor ID: SONY
Product ID: SDX-400C
Barcode:
```

Showing I/O Statistics for a Virtual Tape Drive (vdrive stats)

SYNOPSIS

```
vdrive stats
```

ARGUMENTS

None

RESULT

A successful `vdrive stats` command displays the I/O statistics for the virtual tape drive. On failure, `vdrive stats` returns a non-zero exit status and relevant error message.

Using Shadow Tape APIs

This chapter describes the commands that you use to work with shadow tapes for the NearStore VTL appliance. It consists of the following sections:

- “Listing Shadow Tapes (stape list),” page 127
- “Showing Shadow Tape Details (stape show),” page 127
- “Discovering a Shadow Tape Name or UUID (stape lookup),” page 128

Listing Shadow Tapes (stape list)

SYNOPSIS

```
stape list [stape_id] [count] [start]
```

ARGUMENTS

<i>stape_id</i>	Barcode of the virtual tape
<i>count</i>	Number of tapes to be listed out of <i>n</i> tapes
<i>start</i>	First tape in list of <i>count</i> tapes

RESULT

If *count* or *start* is invalid, the `stape list` command returns a non-zero exit status and relevant error message. Otherwise, the output is one line per shadow tape, in the following format: `<barcode>`.

Showing Shadow Tape Details (stape show)

SYNOPSIS

```
stape show [stape_id] [count] [start]
```

ARGUMENTS

<i>stape_id</i>	Barcode of the virtual tape
<i>count</i>	Number of tapes to be listed out of <i>n</i> tapes
<i>start</i>	First tape in list of <i>count</i> tapes

RESULT

If *count* or *start* is invalid, the `stape show` command returns a non-zero exit status and relevant error message. Otherwise, the output is one line per shadow tape, in the following format:

<Index> <Barcode> <Age>

Discovering a Shadow Tape Name or UUID (stape lookup)

SYNOPSIS

```
stape lookup stape_id|stape_uuid
```

ARGUMENTS

<i>stape_id</i>	Name of the shadow tape
<i>stape_uuid</i>	UUID of the shadow tape

RESULT

If the argument is *stape_id*, the `stape lookup` command displays the UUID of the corresponding shadow tape. If the argument is *stape_uuid*, the `stape lookup` command displays the name of the corresponding shadow tape. If the argument passed is invalid, `stape lookup` returns a non-zero exit status and relevant error message.

Configuring ACSLS

This chapter describes the commands that you use to configure Automated Cartridge System Library Software (ACSL) for the NearStore VTL appliance. It consists of the following sections:

- “Enabling ACSLS,” page 129
- “Setting the ACSLS Server (acsls set),” page 129
- “Showing ACSLS Status (acsls show),” page 130
- “Listing Barcode Ranges (acsls list_barcode_range),” page 131
- “Adding Barcode Ranges to VTL (acsls add_barcode_range),” page 131
- “Removing Barcode Ranges to VTL (acsls remove_barcode_range),” page 132

Enabling ACSLS

The `options set` command has an optional argument, `[use_acsl]`, for enabling ACSLS support. Once ACSLS support is enabled, you cannot disable it. The following text explains the results of the command and argument:

- `[use_acsl]= true` is rejected if there are SCSI libraries added.
- `[use_acsl]= false` is rejected if ACSLS server is configured.

Setting the ACSLS Server (acsls set)

SYNOPSIS

```
acsls set server_host_name [user_id]
```

ARGUMENTS

<i>server_host_name</i>	Fully qualified host name or IP address of the host on which ACSLS is running
<i>user_id</i>	User name with which VTL logs into the ACSLS

RESULT

A successful `acsls set` command adds the ACSLS server with the user-specified configuration. On failure, `acsls set` returns a non-zero exit status and relevant error message.

This command can be used for:

- Setting the ACSLS server parameter for the first time
- Changing the ACSLS server parameters:
 - If the parameters passed to the command are the same as the current values, nothing is done.
 - If the parameters are different, a reboot is required.

Showing ACSLS Status (`acsls show`)

SYNOPSIS

```
acsls show
```

ARGUMENTS

None

RESULT

A successful `acsls show` command shows ACSLS server information, in the following format:

```
<Server Host Name> <User ID> <Communication State> <Server State>
```

On failure, `acsls show` returns the message ACSLS server not configured.

EXAMPLE

```
VTL> acsls show
```

```
Acsls Server Information:-
```

```
Server Host Name: 10.41.84.93
```

```
User ID:
```

```
Communication State: Communication with \
```

```
ACSLs Server OK
```

```
Server State: RUN
```

Listing Barcode Ranges (`acsls list_barcode_range`)

SYNOPSIS

```
acsls list_barcode_range
```

ARGUMENTS

None

RESULT

A successful `acsls list_barcode_range` command lists barcode ranges added to the VTL. On failure, `acsls list_barcode_range` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> acsls list_barcode_range
```

```
0. Barcode Ranges:-
```

```
From: STK000
To: STK999
From: 500000
To: 509999
```

Adding Barcode Ranges to VTL (`acsls add_barcode_range`)

SYNOPSIS

```
acsls add_barcode_range from to [from1] [to1] [from2] [to2] ...
```

ARGUMENTS

<i>from</i>	Barcode range start
<i>to</i>	Barcode range end

RESULT

The user can enter a list of barcode ranges with the correct number of arguments. A successful `acsls add_barcode_range` command adds the tapes in the barcode range to the VTL library. If the barcode ranges are valid, they are added to the ACSLS barcode range list and the system reboots. On failure, `acsls add_barcode_range` returns a non-zero exit status and relevant error message. The `acsls add_barcode_range` command is rejected if an inventory is in progress.

Removing Barcode Ranges to VTL (acsls remove_barcode_range)

SYNOPSIS

```
acsls remove_barcode_range from to
```

ARGUMENTS

<i>from</i>	Barcode range start
<i>to</i>	Barcode range end

RESULT

A successful `acsls remove_barcode_range` command removes the tapes in the barcode range from the VTL library. A user can enter a list of barcode ranges to be removed from the barcode range list. On failure, `acsls remove_barcode_range` returns a non-zero exit status and relevant error message. The `acsls remove_barcode_range` command is rejected if an inventory is in progress.

CHAPTER

35

Using ACS Libraries

This chapter describes the commands that you use to work with the ACS libraries for the NearStore VTL appliance. It consists of the following sections:

- “Listing ACS Libraries (acs list),” page 133
- “Showing ACS Libraries (acs list_acs),” page 134
- “Listing ACS Libraries LSM Details (acs list_lsm),” page 134
- “Listing ACS Library Drive Details (acs list_drive),” page 135
- “Listing ACS Libraries CAP Details (acs list_cap),” page 136
- “Adding ACS Libraries (acs add),” page 136
- “Removing ACS Libraries (acs remove),” page 137
- “Listing Inventory Details of an ACS (acs inventory),” page 137
- “Making a Batch CAP Request (acs batch_cap),” page 138
- “Listing Batch CAP Requests (acs list_batch_cap),” page 138
- “Canceling a Batch CAP Request (acs cancel_batch_cap),” page 139
- “Moving Tapes to CAP (acs eject),” page 139

Listing ACS Libraries (acs list)

SYNOPSIS

```
acs list [acs_id]
```

ARGUMENTS

<i>acs_id</i>	Name or identifier of the ACS library
---------------	---------------------------------------

RESULT

A successful `acs list` command displays configured ACS libraries. If `acs_id` is passed, then `acs list` displays that library if it is configured. On failure, `acs list` returns a non-zero exit status and relevant error message.

Showing ACS Libraries (acs list_acs)

SYNOPSIS

```
acs list_acs [acs_id]
```

ARGUMENTS

<code>acs_id</code>	Name or identifier of the ACS library
---------------------	---------------------------------------

RESULT

A successful `acs list_acs` command displays details (nickname, ACS ID, ACS state, drive count, cap count, LSM count) of all the ACS libraries. If `acs_id` is passed, then `acs list_acs` displays details for that library. On failure, `acs list_acs` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> acs list_acs
0. Acs Library Information:-
                                Nickname: acs_lib1
                                ACS ID: 0
                                ACS State: enabled
                                LSM Count: 1
                                Drive Count: 2
                                CAP Count: 1
```

Listing ACS Libraries LSM Details (acs list_lsm)

SYNOPSIS

```
acs list_lsm [acs_id]
```

ARGUMENTS

<code>acs_id</code>	Name or identifier of the ACS library
---------------------	---------------------------------------

RESULT

A successful `acs list_lsm` command displays LSM details (nickname, ACS ID, ACS state, LSM ID, LSM date, serial number, model) of all the ACS libraries. If `acs_id` is passed, then `acs list_lsm` displays details for that library. On failure, `acs list_lsm` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> acs list_acs 0
```

```
Acs Library Information:-
```

```

Nickname: acs_lib1
ACS ID: 0
ACS State: enabled
LSM Count: 1
Drive Count: 2
CAP Count: 1
```

Listing ACS Library Drive Details (acs list_drive)

SYNOPSIS

```
acs list_drive [acs_id]
```

ARGUMENTS

<i>acs_id</i>	Name or identifier of the ACS library
---------------	---------------------------------------

RESULT

A successful `acs list_drive` command displays drive details (nickname, ACS ID, ACS State, drive ID, serial number, model, drive state, and barcode) of all the ACS libraries. If *acs_id* is passed, then `acs list_drive` displays details for that library. On failure, `acs list_drive` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> acs list_drive 0
```

```
0. Acs Library Drive Information:-
```

```

Nickname: acs_lib1
ACS ID: 0
ACS State: enabled
Drive ID: 0,0,1,1
Drive State: ONLINE
Drive Serial Number: HU106229HJ
Drive Model: N/A
Barcode:
Drive ID: 0,0,1,2
Drive State: ONLINE
Drive Serial Number: HU10634H0Y
Drive Model: N/A
Barcode:
```

Listing ACS Libraries CAP Details (acs list_cap)

SYNOPSIS

`acs list_cap [acs_id]`

ARGUMENTS

<i>acs_id</i>	Name or identifier of the ACS library
---------------	---------------------------------------

RESULT

A successful `acs list_cap` command displays CAP details (nickname, ACS ID, ACS state, CAP ID, CAP priority, CAP size, CAP mode, CAP state) of all the ACS libraries. If *acs_id* is passed, then `acs list_cap` displays details for that library. On failure, `acs list_cap` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> acs list_cap 0
0. Acs Library CAP Information:-
                                Nickname: acs_lib1
                                ACS ID: 0
                                ACS State: enabled
                                CAP ID: 0,0,0
                                CAP Priority: 0
                                CAP Size: 5
                                CAP Mode: MODE_MANUAL
                                CAP State: ONLINE
```

Adding ACS Libraries (acs add)

SYNOPSIS

`acs add acs_id nickname`

ARGUMENTS

<i>acs_id</i>	Name or identifier of the ACS library
<i>nickname</i>	Nickname to identify the library

RESULT

A successful `acs add` command adds the ACS library to the appliance configuration. If the parameter passed is invalid, `acs add` returns a non-zero exit status and relevant error message.

Removing ACS Libraries (**acs remove**)

SYNOPSIS

```
acs remove acs_id
```

ARGUMENTS

<i>acs_id</i>	ID of the ACS library
---------------	-----------------------

RESULT

A successful `acs remove` command removes the ACS library specified in the argument. If the argument is invalid, `acs remove` returns a non-zero exit status and relevant error message.

Listing Inventory Details of an ACS (**acs inventory**)

SYNOPSIS

```
acs inventory acs_id [start][count]
```

ARGUMENTS

<i>acs_id</i>	ID of the ACS library.
<i>start</i>	Starting tape count from where the listing should be done.
<i>count</i>	Number of entries to be listed. The default is 50 (only if <i>start</i> is passed as a parameter).

RESULT

If the *acs_id* is valid, the `acs inventory` command prints details for that ACS library, in the following format:

```
<Barcode> <Location ID> <Date Cloned> <Date Exported> <Virtual Library>
<Date Virtual Tape Written> <In Sync> <Status>
```

If the argument passed is invalid, `acs inventory` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> acs inventory 0
```

```
Acs Inventory Details:-
```

```

                Barcode: 500000
            Location ID: 0,0,1,3,2
            Date Cloned: N/A
            Date Exported: N/A
        Virtual Library: N/A
    Date Virtual Tape Written: N/A
```

In Sync: N/A

Status: NEW

Making a Batch CAP Request (`acs batch_cap`)

SYNOPSIS

```
acs batch_cap acs_id barcode
```

ARGUMENTS

<i>acs_id</i>	ID of the ACS library
<i>barcode</i>	Physical tape label

RESULT

A successful `acs batch_cap` command moves the barcode to a batch CAP request list. On failure, `acs batch_cap` returns a non-zero exit status and relevant error message.

Listing Batch CAP Requests (`acs list_batch_cap`)

SYNOPSIS

```
acs list_batch_cap acs_id [start] [count]
```

ARGUMENTS

<i>acs_id</i>	ID of the ACS library.
<i>start</i>	Starting tape count from where the listing should be done.
<i>count</i>	Number of entries to be listed. The default is 50 (only if <i>start</i> is passed as a parameter).

RESULT

A successful `acs list_batch_cap` command lists the barcode in a batch cap request list. On failure, `acs list_batch_cap` returns a non-zero exit status and relevant error message.

EXAMPLE

```
VTL> acs list_batch_cap 0
```

```
Acs Batch Cap Details:-
```

```
Barcode: 500019
Location ID: 0,0,1,3,4
Date Cloned: N/A
Date Exported: N/A
Virtual Library: N/A
```

Date Virtual Tape Written: N/A
In Sync: N/A
Status: Waiting to be processed

Canceling a Batch CAP Request (`acs cancel_batch_cap`)

SYNOPSIS

```
acs cancel_batch_cap acs_id barcode
```

ARGUMENTS

<i>acs_id</i>	ID of the ACS library
<i>barcode</i>	Physical tape label

RESULT

A successful `acs cancel_batch_cap` command removes the barcode from a batch cap request list. On failure, `acs cancel_batch_cap` returns a non-zero exit status and relevant error message.

Moving Tapes to CAP (`acs eject`)

SYNOPSIS

```
acs eject acs_id cap_id barcode
```

ARGUMENTS

<i>acs_id</i>	ID of the ACS library
<i>cap_id</i>	CAP ID to which tapes to be moved
<i>barcode</i>	Physical tape to be moved to CAP

RESULT

A successful `acs eject` command moves the barcode to CAP. On failure, `acs eject` returns a non-zero exit status and relevant error message.

36

Licensing

This chapter describes the commands that you use to manage licensing for the NearStore VTL appliance. It consists of the following sections:

- “Listing Installed Licenses (license list),” page 141
- “Showing Installed Licenses (license show),” page 141
- “Adding a License (license add),” page 142

Listing Installed Licenses (license list)

SYNOPSIS

```
license list license_id
```

ARGUMENTS

<i>license_id</i>	Name or identifier of the license installed
-------------------	---

RESULT

A successful `license list` command lists all the installed licenses. On failure, `license list` returns a non-zero exit status and relevant error message.

Showing Installed Licenses (license show)

SYNOPSIS

```
license show license_id
```

ARGUMENTS

<code>license_id</code>	Name or identifier of the license installed
-------------------------	---

RESULT

A successful `license show` command displays details of all the installed licenses. If `license_id` is passed, then `license show` displays details for that license id. On failure, `license show` returns a non-zero exit status and relevant error message.

Adding a License (license add)

SYNOPSIS

`license add license_key`

ARGUMENTS

<code>license_key</code>	14-character license key
--------------------------	--------------------------

RESULT

A successful `license add` command adds the license into appliance configuration. On failure, `license add` returns a non-zero exit status and relevant error message.

Upgrading Firmware

This chapter describes the commands that you use to upgrade VTL, shelf controller, and disk firmware for the NearStore VTL appliance.

Some commonly used shelf controller and disk (but not VTL) firmware images are stored on VTL appliances when they are shipped. You can use a CLI command to install one or more of these prepackaged image files.

In addition, VTL, shelf controller, and disk firmware upgrade files are available for download on the NOW NetApp on the Web® site at <http://now.netapp.com/>. You can download a file, upload it to the VTL appliance through an FTP client, and then use a CLI command to install it.

After you download a VTL, shelf controller, or disk firmware file from the NOW site, follow this procedure to upload the file to the VTL appliance:

1. On your client computer, from the directory in which you stored the downloaded firmware file, use an FTP client to connect to the VTL appliance:

```
FTP> open VTL_host_name|VTL_ip_address
```

Enter the same user name and password that you use to access the VTL CLI or graphical user interface (GUI).

2. Transfer the downloaded firmware file to the VTL appliance:

```
FTP> put firmware_file_name
```

3. Make sure that the file was uploaded successfully, as shown:

```
FTP> ls
```

The file's name and size should be displayed. You are now ready to use a CLI command to install the firmware image.

Note

Once you begin to install a firmware image that was downloaded, you will not be able to see that image file on the VTL appliance through the `FTP ls` command. When installation is complete, the uploaded firmware image file is deleted from the FTP space. If you have uploaded the wrong file, the only way to delete it is to reboot the appliance.

If you upload a file larger than 50 MB, an error occurs, and the file is not uploaded.

This chapter consists of the following sections:

- “Installing VTL Firmware (appliance install),” page 144
- “Listing Shelf Controller Firmware (shelffirmware list),” page 145
- “Installing Downloaded Shelf Controller Firmware on Specific Shelves (shelffirmware install),” page 145
- “Installing Downloaded Shelf Controller Firmware on All Shelves (shelffirmware install *image_name* all),” page 145
- “Installing Prepackaged Shelf Controller Firmware (shelffirmware install all),” page 146
- “Viewing Shelf Controller Firmware Status (shelffirmware status),” page 146
- “Listing Disk Firmware (diskfirmware list),” page 147
- “Installing Disk Firmware on Disks of a Specific Model (diskfirmware install),” page 147
- “Installing Downloaded Firmware on All Disks (diskfirmware install all ftp),” page 148
- “Installing Prepackaged Firmware on All Disks (diskfirmware install all default),” page 148
- “Viewing Disk Firmware Status (diskfirmware status),” page 148

Installing VTL Firmware (appliance install)

SYNOPSIS

`appliance install image_name`

ARGUMENTS

<i>image_name</i>	File name of the VTL image
-------------------	----------------------------

RESULT

A successful `appliance install` command installs a new VTL firmware image on your NearStore VTL appliance and returns the following message:

VTL Firmware installed successfully. Please reboot the appliance.

On failure, the CLI displays one of the following messages:

ERROR: Cannot downgrade product.

ERROR: Firmware Image is not found.

Listing Shelf Controller Firmware (shelffirmware list)

SYNOPSIS

```
shelffirmware list
```

ARGUMENTS

None

RESULT

A successful `shelffirmware list` command displays information about shelves attached to the VTL and the current firmware revision.

Installing Downloaded Shelf Controller Firmware on Specific Shelves (shelffirmware install)

SYNOPSIS

```
shelffirmware install image_name shelf_id0 shelf_id1
```

ARGUMENTS

<i>image_name</i>	File name of the shelf firmware image.
<i>shelf_id#</i>	ID of the shelf being upgraded. You can specify from one to eight shelves.

RESULT

A successful `shelffirmware install` command installs, on specific shelves, the shelf controller firmware that you downloaded from the NOW site and uploaded to the appliance. On failure, `shelffirmware install` returns an error message.

Installing Downloaded Shelf Controller Firmware on All Shelves (shelffirmware install image_name all)

SYNOPSIS

```
shelffirmware install image_name all
```

ARGUMENTS

<i>image_name</i>	File name of the shelf firmware image
-------------------	---------------------------------------

RESULT

A successful `shelffirmware install image_name all` command installs, on all shelves, the new shelf controller firmware that you downloaded from the NOW site and uploaded to the appliance. On failure, `shelffirmware install image_name all` returns an error message.

Installing Prepackaged Shelf Controller Firmware (shelffirmware install all)

SYNOPSIS

```
shelffirmware install all
```

ARGUMENTS

None

RESULT

A successful `shelffirmware install all` command installs prepackaged shelf controller firmware on all shelves. On failure, `shelffirmware install all` returns an error message.

Note

If you want to install prepackaged shelf controller firmware on some shelves but not others, you must download the file from the NOW site, upload it to your VTL appliance, and use the `shelffirmware install` command, which allows you to specify shelves.

Viewing Shelf Controller Firmware Status (shelffirmware status)

SYNOPSIS

```
shelffirmware status
```

ARGUMENTS

None

RESULT

A successful `shelffirmware status` command shows the status of the shelves. On failure, `shelffirmware status` returns an error message.

Listing Disk Firmware (diskfirmware list)

SYNOPSIS

```
diskfirmware list
```

ARGUMENTS

None

RESULT

A successful `diskfirmware list` command displays all models of disks present on the shelf or shelves attached to the VTL and the number of disks that belong to each model. The command also lists the recommended firmware revision for each model and the number of mismatched disks, if present. On failure, `diskfirmware list` returns an error message.

Installing Disk Firmware on Disks of a Specific Model (diskfirmware install)

SYNOPSIS

```
diskfirmware install model fw_rev [file1] [file2]
```

ARGUMENTS

<i>model</i>	Model displayed by the <code>diskfirmware list</code> command.
<i>fw_rev</i>	Recommended revision displayed by the <code>diskfirmware list</code> command. If the revision is N/A, no upgrade is recommended for the disk. If you want to install a non-recommended version of firmware, specify it.
<i>file1</i>	Name of the upgrade binary file.
<i>file2</i>	Name of the upgrade control file, which some vendors require in addition to the binary file. <i>file1</i> and <i>file2</i> (if required) can be specified in any order.

RESULT

A successful `diskfirmware install` command installs firmware, either the prepackaged VTL copy or the version from the NOW site, on disks of a specific model. Enter the *file1* and *file2* arguments *only* if you are installing a firmware image that you downloaded. Omitting the *file1* and *file2* arguments installs a prepackaged version of the disk firmware. If the arguments are omitted and no prepackaged version of the firmware is stored on the VTL, the system displays a message telling you to upload a file to the appliance. On failure, `diskfirmware install` returns an error message.

Installing Downloaded Firmware on All Disks (`diskfirmware install all ftp`)

SYNOPSIS

```
diskfirmware install all ftp
```

ARGUMENTS

None

RESULT

The `diskfirmware install all ftp` command looks at all downloaded disk firmware upgrade files stored on the VTL appliance, picks the ones required for each disk model, and installs the upgrade on all the disks. A successful command returns the installation status for each model. On failure, `diskfirmware install all ftp` returns an error message.

Installing Prepackaged Firmware on All Disks (`diskfirmware install all default`)

SYNOPSIS

```
diskfirmware install all default
```

ARGUMENTS

None

RESULT

The `diskfirmware install all default` command looks at all prepackaged disk firmware upgrade files that shipped with the VTL appliance, picks the ones required for each disk model, and installs the upgrade on all the disks. A successful command returns the installation status for each model. On failure, `diskfirmware install all default` returns an error message.

Viewing Disk Firmware Status (`diskfirmware status`)

SYNOPSIS

```
diskfirmware status
```

ARGUMENTS

None

RESULT

A successful `diskfirmware status` shows detailed status information for each model of disk undergoing an upgrade. The display is organized by disk model, and lists the recommended firmware version, the total number of disks selected for upgrade, the currently active upgrades, the number of upgrades that have failed, and the number of pending upgrades. If the firmware installation fails, the `diskfirmware status` command lists the information for each disk that has failed or is in pending state.

If any of the upgrades fail, you cannot ignore the failure in the CLI as you can through the GUI. You must access the VTL's GUI, and fix or ignore the problem through the appropriate resolve link on the Admin page. To retry the failed disk, you must again execute the `diskfirmware install` command through the CLI. If you used a downloaded firmware image instead of a prepackaged firmware image, you must upload the file again to the VTL appliance.